UNIVERSITY OF CALIFORNIA SAN DIEGO

Tension Optimization Control of the Cable-driven Boat motion simulator, and the Design of Pipeline Exploration and Inspection Robot

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Engineering Sciences (Mechanical Engineering)

by

Joon Young Jang

Committee in charge:

Professor Thomas Bewley, Chair
Professor Maurício de Oliveira
Professor Sylvia Herbert
Professor Falko Kuester

2024

The Dissertation of Joon Young Jang is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2024

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to acknowledge Professor Thomas Bewley's invaluable support as the chair of my committee. Through numerous drafts and many long nights, his guidance his guidance has been indispensable. Additionally, I extend my appreciation to my colleagues at the Coordinated Robotics Lab. Without their support, my research undoubtedly would have taken significantly longer. Their assistance has been immeasurable.

Support for this work was provided by the Republic of Korea(ROK) Navy Sponsored Education Program. I am grateful for their support for this work. A heartfelt thank you goes out to my coworkers at the ROK Navy headquarters for their assistance in facilitating my research endeavors here.

A sincere and heartfelt thank you goes out to my family, whose unwavering support and motivational encouragement have been a constant source of strength for me. I am also deeply grateful to both Long Sensei and Lee Sensei, who have served as guiding lights in my life. Additionally, I extend my appreciation to all members of the Iaido family and my colleagues for their continuous encouragement and support.

Chapter 1, in part, is a reprint of the material as it appears in the International Conference on Robotics Systems and Automation Engineering(RSAE) 2021, J. Jang and T. Bewley, "Tension optimization of the 6-DOF cable-driven boat motion simulator". The dissertation author was the primary investigator and author of this paper.

Chapter 2, in full, is a reprint of the material as it appears in IEEE International Conference on Robotics and Automation Engineering(ICRAE) 2022, J. Jang, M. Zhao, and T. Bewley, "A Control Lyapunov Function-Based Quadratic Program for the Cable-Driven Boat Motion Simulator". The dissertation author was the primary investigator and author of this paper.

Chapter 3, in full, is a reprint of the material as it appears in IEEE International Conference on Control and Robotics(ICCR) 2023, J. Jang, and T. Bewley, "Workspace Analysis for Cable-Driven Parallel Robot based Boat Motion Simulator". The dissertation author was the primary investigator and author of this paper.

Chapter 4, in part, is currently being prepared for submission for publication of the material, J. Jang, T. Bewley. The dissertation author was the primary investigator and author of this paper.

Chapter 5, in full, is a reprint of the material as it appears in IEEE 8th International Conference on Control and Robotics Engineering(ICCRE), 2023, J. Jang, M. A. M. Macias and T. Bewley, "Camera Image Based Moving Platform Rotation Estimation for Quadrotor Landing". The dissertation author was the primary investigator and author of this paper.

Chapter 6, in part, has been submitted for publication of the material as it may appear in IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS) 2024, J. Jang, Z. Hooker, T. Wan, P. Tejeda, and T. Bewley, "Design and Characterization of the Torsion Spring-Motor Integrated Series Elastic Actuator". The dissertation author was the primary investigator and author of this paper.

Chapter 6, in part, has been submitted for publication of the material as it may appear in IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS) 2024, J. Jang, T. Wan, Z. Hooker, P. Tejeda, and T. Bewley, "A Novel Design of 4-Omniwheel Robot for Interior Exploration and Inspection of Pipe Network". The dissertation author was the primary investigator and author of this paper.

VITA

2009	B. S. in Mechanical Engineering and Naval Architecture, Republic of Korea Naval Academy
2009–Current	Navy Officer, Republic of Korea Navy
2020	M. S. in Engineering Sciences (Mechanical Engineering), University of California San Diego
2020–2021	Graduate Teaching Assistant, University of California San Diego
2024	Ph. D. in Engineering Sciences (Mechanical Engineering), University of California San Diego

PUBLICATIONS

J. Jang and T. Bewley, "Tension optimization of the 6-DOF cable-driven boat motion simulator," in 2021 the 3rd International Conference on Robotics Systems and Automation Engineering (RSAE), May 2021.

J. Jang, M. Zhao, and T. Bewley, "A Control Lyapunov Function-Based Quadratic Program for the Cable-Driven Boat Motion Simulator," in 2022 7th International Conference on Robotics and Automation Engineering (ICRAE), pp. 18-24, IEEE, 2022.

J. Jang, M. A. M. Macias and T. Bewley, "Camera Image Based Moving Platform Rotation Estimation for Quadrotor Landing," in 2023 8th International Conference on Control and Robotics Engineering (ICCRE), pp. 143-149, IEEE, 2023.

T. Bewley, J. Jang, T. Wan, Z. Hooker, P. Tejeda, "A new class of stable, spring-loaded, planar, roll-controlled, highly maneuverable robots for interior exploration of networks of pipes and ducts," May 8 2023, University of California San Diego, Patent Case SD2023-185.

J. Jang, and T. Bewley, "Workspace Analysis for Cable-Driven Parallel Robot based Boat Motion Simulator," in 2023 the 5th International Conference on Control and Robotics(ICCR), IEEE, Nov 2023.

J. Jang, Z. Hooker, T. Wan, P. Tejeda, and T. Bewley, "Design and Characterization of the Torsion Spring-Motor Integrated Series Elastic Actuator," IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS), 2024, Submitted.

J. Jang, Z. Hooker, T. Wan, P. Tejeda, T. Bewley, "A Novel Design of 4-Omniwheel Robot for Interior Exploration and Inspection of Pipe Networks," IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS), 2024, Submitted.

ABSTRACT OF THE DISSERTATION

Tension Optimization Control of the Cable-driven Boat motion simulator, and the Design of Pipeline Exploration and Inspection Robot

by

Joon Young Jang

Doctor of Philosophy in Engineering Sciences (Mechanical Engineering)

University of California San Diego, 2024

Professor Thomas Bewley, Chair

This study consists of two parts, and the first part is the Cable-driven boat motion simulator. In naval operations, vertical take-off and landing (VTOL) drones are widely used as cost-effective assets that are easy to maneuver, enhancing the reconnaissance and surveillance capabilities of ships. Despite their advantages, these drones have limited operational time and need to return to the base platform. To address the challenge of developing drone landing systems, conducting experiments at sea poses significant costs and risks. This dissertation proposes the utilization of a ground-based motion simulator that mimics the 6-DOF motion of a boat. The

simulator aims to replicate the sea environment up to sea state 6 on the Doublas scale. To achieve motion control and optimization of the cable-driven boat motion simulator, a Control Lyapunov function-based Quadratic Program (CLF-QP) control method was studied. Given the importance of cable tension in cable-driven robots, the CLF-QP control method facilitates platform movement in the desired 6-DOF motions while ensuring that cable tensions remain within specified limits. In addition to the investigation into designing a boat motion simulator, a related research project focused on camera image-based pose estimation for drone landing. This study is preliminary research on drone landing control and specifically addresses the 3D rotation estimation of a moving platform using 2D images captured by a camera. The research assumes the presence of a circular pattern marker on the flight deck of a ship, with the drone hovering at the center of the platform. Equipped with a camera, the drone captures 2D images of the markers and measures the distance between them. As the platform rotates, the circular pattern of markers transforms into an ellipse. The study derives the ellipse equation from the marker positions on the ellipse, utilizing geometry and trigonometric functions to determine the platform's rotation sequence. By estimating the platform's rotation, the drone can effectively determine the optimal timing for the platform to rotate within the drone's capacity to land safely. The proposed method was validated through simulation and hardware testing, with a subsequent discussion on the platform state, and tension tracking errors obtained by a motion capture system and load-cell tension sensors.

The second part of this study is the Pipeline exploration and inspection robot. The research introduces a new robot design specifically tailored for inspecting, exploring, and mapping intricate pipe networks featuring multiple curves and joints. The proposed robot, named Model S4, stands out for its simplicity, high maneuverability, and static stability, achieved through the incorporation of four omniwheels arranged in a single plane at the robot center. The robot is equipped with high-torque motors that effectively control roll motions. The robot employs a spring-servo motor integrated series elastic actuator to enhance stability. This mechanism ensures that the robot maintains three or more points of contact with the opposite sidewalls of the pipe,

causing the robot's center plane to consistently align with the centerline of the pipe throughout its operation. This design promotes stable travel within the pipe at any orientation relative to gravity, without the need for constant feedback. Additionally, the robot features mechanisms designed for efficient navigation through pipe curves and T-joints, a capability validated through hardware testing focusing on roll control and maneuvering through pipe curves and joints.

The primary contributions of this dissertation are as follows:

1. Control Lyapunov Function-based Quadratic Program (CLF-QP) control for motion control and tension optimization of the cable-driven boat motion simulator.

2. The workspace Analysis for Parameter Optimization of the cable-driven boat motion simulator utilizing static analysis, cable-to-cable and cable-to-platform interference detection algorithm.

3. Hardware model building and testing to verify the performance of CLF-QP control with the discussion of platform pose and tension tracking errors measured by a motion capture system and tension sensors.

4. The simulation and hardware experimental validation of a camera image-based platform rotation estimation.

5. A novel design of a robot for interior exploration and inspection of pipe networks embedding the torsion spring and servo motor integrated series elastic actuator.

Introduction

Vertical take-off and landing (VTOL) drones are widely utilized as cost-effective assets and are easy to maneuver. In naval operations, these drones play a crucial role in extending the reconnaissance and surveillance capabilities of ships. However, executing landing at sea presents a significant challenge, especially considering the dynamic movement of landing platforms caused by waves and winds. Thus, the development of sea landing controllers and the validation of drone landing capabilities benefit from the use of a boat motion simulator. The boat motion simulator proves invaluable for conducting land-based tests, mitigating cost and risk associated with at-sea experiments. While the Stewart platform mechanism offers a 6-degree-of-freedom (6-DOF) capability, its intricate mechanical system limits the workspace range [1]. The 3-PSR Parallel mechanism was introduced for replicating boat motion [2], but its upper structure hinders drone landing capabilities. Inspired by Max Planck's CableRobot Simulator [3], this study proposes a cable-driven parallel robot(CDPR) mechanism based simulator. CDPRs, functioning as tensegrity systems with multiple bars and cables, offer a unique approach. Analyzing the statics and dynamics of the tensegrity system proves to be a valuable method for understanding the forces acting on both bars and cables [4]. Tensegrity analysis has been successfully employed in stabilizing multi-tethered balloons, ensuring optimal design to maintain cable tension in the presence of disturbances such as winds [5]. This cable-driven simulator aims to provide an effective solution for simulating boat motion, contributing to advancements in drone landing capabilities.

By manipulating the length of cables connected to the moving platform, precise control over the platform's motion in all 6-DOF can be achieved with high acceleration and accuracy

[6]. Previous research has addressed various challenges associated with cable-driven systems, including workspace analysis [7][8] and the development of cable collision detection algorithms [9]. The advantage of cable-driven systems lies in the simplicity of their kinematics and broad applicability. However, it is important to note a key characteristic: cables can only pull an object and cannot push it. Consequently, maintaining positive cable tensions is essential for robust and stable motion simulations. Insufficient cable tensions may lead to slack in cables, risking system failure, while excessive tensions can potentially damage the mechanical structure. A cable-driven system with more cables than its degrees of freedom is termed an underdetermined system, wherein cable tension solutions are not unique. Cable length at every state of the platform can be calculated by inverse kinematics [10], and optimal tension solutions also must be considered simultaneously. Some studies have utilized the null space of the static equation solution to determine feasible cable tension solutions [6][11]. However, optimizing tension must be performed dynamically within the control system for a moving platform. While proportional–derivative controllers (PD) have been employed to control cable-driven robot motions [12][13][14], these studies have primarily focused on accurate motion control rather than tension optimization. In addition, the derivative term in a PD controller can amplify noise in the system, leading to actuator oscillations [15]. Controllers for cable-driven robots must effectively manage platform motions while ensuring that cable tensions remain within the desired range. Given the numerous feasible tension solutions in underdetermined CDPR at each equilibrium state [16], optimizing cable tensions within specified boundaries becomes a crucial aspect of their control.

In our prior research on the control and tension optimization of the cable-driven boat motion simulator, we proposed the use of a Linear Matrix Inequality (LMI)-based controller [17]. We constructed a simulation model grounded in the kinematics and dynamics of the tensegrity system. The simulation outcomes revealed that the LMI controller effectively governed the motions of the moving platform with precision, and the standard deviation method optimized cable tensions within the desired boundaries. However, we encountered challenges related to

the inefficiency of the linearization process of tensegrity system dynamics and the intricacy of cable tension optimization algorithms. Thus, the computational demands became burdensome, particularly when applied to control an experimental hardware model. The intricacies in these processes posed challenges, prompting further refinement to streamline the computational aspects of controlling the cable-driven boat motion simulator.

Chapters 1 to 4 delve into the exploration of the CDPR boat motion simulator. Chapter 1 serves as an introduction, laying the foundation with a comprehensive overview of the basic theory behind the CDPR mechanism. This includes the discussion of kinematics, static equilibrium, and dynamics, providing a solid understanding of the fundamental principles governing the motion of the system. Moreover, Chapter 1 includes the analysis of CDPRs by leveraging Tensegrity theories. This approach adds a unique perspective to the study, exploring the interplay between cable-driven mechanisms and principles derived from tensegrity, enriching the theoretical framework.

In Chapter 2, we present a Control Lyapunov Function-based Quadratic Program (CLF-QP) controller design method. This approach is grounded in the nonlinear controller design method, which is widely applicable to closed-loop feedback control systems [18]. The versatility of CLF controllers makes them practical for diverse applications, even those with fast sample rates [19]. Through the selection of a suitable CLF that satisfies specific conditions, it becomes possible to determine the optimal input that stabilizes a system. The presence of a CLF indicates the stabilizability of a system, and when integrated with the Quadratic Program (QP) method, the CLF approach preserves the performance characteristics defined by a CLF while adhering to user-assigned constraints, such as tension boundaries [20].

Chapter 3 discusses the workspace analysis for parameter optimization of the able-driven moat motion simulator. The replication of the desired boat motion can be achieved by the proper optimization of design parameters that ensure an ample workspace for the simulator. This chapter focuses on analyzing the workspace using three design parameters: height, width, and cable attachment points. Examining the workspace under various parameter configurations aims

to optimize the design and ensure that the simulator provides a sufficient range of motion to replicate complex boat motions. To assess the workspace, we developed a simulation model and an algorithm incorporating three methods: static equilibrium analysis, cable-to-cable, and cable-to-platform interference detection algorithm. Through these methods, the research aims to provide insights into achieving an optimized design for the simulator.

Chapter 4 introduces an examination of a cable-driven boat motion simulator through both simulation and hardware models, aiming to validate the performance of the Control Lyapunov Function-based Quadratic Program (CLF-QP) control method. In both models, the configuration involves eight cables and 6-DOF with a moving platform, and both models' sizes are equivalent. The simulation model test is designed to observe position and rotation tracking errors. Additionally, the simulation evaluates the efficacy of tension optimization while simulating the desired boat motion. This simulation result provides insights allowing for a thorough assessment of the CLF-QP control method. The hardware model incorporates eight tension sensors connected between cables and the platform to measure cable tensions. Each motor is equipped with an encoder sensor and a roller for cable winding. The PID controller was applied to the control hardware model, leveraging the measured tension data as the system's feedback. The hardware model test, therefore, focuses on observing the outcomes of the CLF-QP-based tension optimization controller in a real-world setting.

Chapter 5 proposes a camera image-based method for estimating the rotation of a moving platform, with a specific focus on its application in drone landing scenarios. The motivation for this research lies in the need to safely recover aerial vehicles, particularly Vertical Take Off and Landing Unmanned Aerial Vehicles (VTOL UAVs) such as quadrotors. Given the limited operation time of these vehicles due to battery constraints, landing on moving platforms safely becomes crucial for extending their operational capabilities. While previous studies have explored approaches such as tethering power cables to extend quadrotor operation time [2], the requirement for returning to a base platform for maintenance or further operations remains. The challenge becomes even more complex in naval operations, where platforms like navy ships

undergo 6-DOF motion due to winds and waves. Drawing inspiration from the landing process of navy helicopters on ship flight decks [21], the chapter suggests the incorporation of rotation estimation in the quadrotor landing process. In the video, a navy helo pilot tries to land on a navy ship in a rough sea. A pilot waits for the platform motion to be stable enough and lands on the flight deck.

The proposed method involves using a camera on the quadrotor to capture 2D images and recognize markers on the flight deck. The camera acknowledges circular images, measures the pixel position of the circles [22]. The camera image sensor center is aligned with the circular object's center [23]. As the object rotates, the contour of the image changes, resulting in a larger eccentricity. By determining the ellipse equation from marker positions and analyzing the geometric characteristics of the ellipse, rotation estimation can be achieved [25]. A simulation model was developed to validate the proposed method, replicating ship motion and circular pattern markers to imitate conic section camera images. The simulation demonstrated that as the platform rotated, circular pattern markers formed an ellipse, and the rotation estimation algorithm accurately evaluated the rotation angles of the deck. Furthermore, a hardware test was conducted to verify the performance of the proposed method.

Chapter 6 introduces a new design of a robot tailored for inspections, exploration, and mapping within intricate pipe networks featuring multiple curves and joints. The novel pipe inspection robot, Model S4, stands out with its configuration of four omniwheels arranged in a single plane at the robot center. This design choice emphasizes simplicity, high maneuverability, and static stability, positioning the robot as an efficient tool for navigating complex pipe environments. Model S4 is equipped with high-torque gear motors capable of controlling roll motions. The robot incorporates a spring-servo motor integrated series elastic actuator(SEA) to enhance stability. SEA maintains three or more points of contact with the pipe walls, ensuring that the robot's center plane consistently aligns with the centerline of the pipe throughout its operation. This design feature enables stable travel within the pipe at any orientation relative to gravity, eliminating the need for constant feedback mechanisms. The robot is equipped with mechanisms

specifically designed to navigate 4-inch pipe curves and T-joints effectively. The effectiveness of these features is validated through hardware testing, focusing on roll control and the robot's ability to maneuver through pipe curves and joints. The chapter illustrates the implementation of the proposed design, highlighting its potential for practical applications in the exploration and inspection of complex pipe networks.

This dissertation is structured as follows. Chapter 1 provides an in-depth exploration of the kinematics and dynamics of cable-driven robots. Chapter 2 describes the design and workspace optimization of a boat motion simulator model leveraging static analysis and interference detection algorithm. Chapter 3 focuses on control and tension optimization, introducing the Control Lyapunov Function-based Quadratic Program (CLF-QP) controller as a key element in achieving to replicate the desired motion. Chapter 4 outlines the experimental test setup and validation procedures, providing insights into the practical implementation and performance of the proposed models and controllers. Chapter 5 introduces a novel approach to quadrotor landing through camera image-based rotation estimation. This chapter includes simulation results and hardware test outcomes. Chapter 6 discusses the development and features of a new pipe network inspection robot, Model S4, emphasizing its design principles and practical applications. Chapter 7 outlines future avenues for research, presenting potential directions and areas of expansion in the field. Finally, Chapter 8 offers a comprehensive summary of the conclusions drawn from the dissertation, highlighting key findings and contributions made throughout the study.

Chapter 1

Kinematics and Dynamics of Cable-driven Robots

1.1 Kinematics and Static Equilibrium

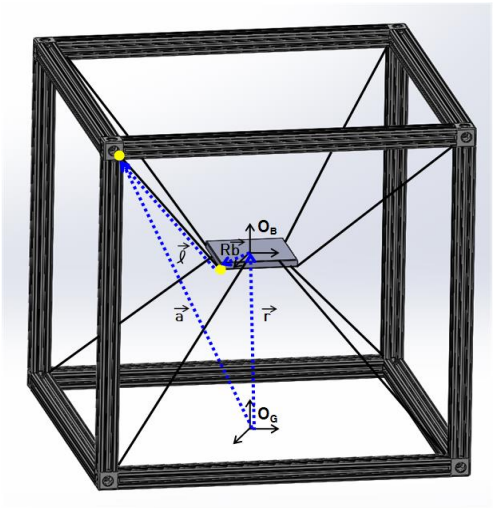


Figure 1.1. The cable-driven boat motion simulator model diagram. A flat-plate cuboid platform is connected with eight cables, and motors adjust cable lengths. The lower and the upper cables are cross-connected to the platform, enhancing the simulator's capability for rotational motions.

The kinematics of Cable-Driven Parallel Robots (CDPRs) can be divided into two parts: Forward Kinematics and Inverse Kinematics. Forward Kinematics involves the calculation of the platform's position based on the given cable lengths. This process allows for the determination of the spatial coordinates and orientation of the platform in response to the adjustments in cable lengths. Conversely, Inverse Kinematics works in the opposite direction. It calculates the cable

length vectors and structure matrix required to achieve a particular platform position. This involves determining the cable tensions necessary to position the platform in a specific manner. In summary, Forward Kinematics is concerned with predicting the platform's position given cable lengths, while Inverse Kinematics involves calculating cable lengths and tensions needed to achieve a desired platform position.

In Figure 1.1, the configuration of the cable-driven boat motion simulator is depicted. The global frame O_G is located at the center of the ground, while the body frame O_B is positioned in the middle of the moving platform, at the platform's center of gravity. The vector $\vec{a}$ represents the pulley position, where $i = \{1, \cdots, n\}$ denotes the i-th cable and n is the total number of cables. The vector $\vec{p} = [x\ y\ z]^T$ denotes the position of the moving platform in the global frame O_G, and $\vec{b}$ represents the cable attachment point vector in the body frame O_B. R is the platform's rotation matrix in the global frame O_G. The cable vector $\vec{l}$ can be determined using inverse kinematics equations, allowing for the calculation of cable lengths based on the desired position and orientation of the moving platform as follows,

$$\vec{l_i} = \vec{a_i} - R\vec{b_i} - \vec{p}, \quad (i = 1, \ldots, n). \tag{1.1}$$

where, $\vec{l}$ is the cable vector, $\vec{a}$ is the pulley position vector in the global from O_G, R is the platform rotation matrix, $\vec{b}$ is the cable attachment point vector in the body frame O_B, and $\vec{p}$ is the position vector of a platform.

Forward kinematics is employed to calculate the position from given cable lengths. The equation $v_i(\vec{l}, \vec{p}, R)$ represents a non-linear relationship involving the i-th cable length, the position($\vec{p}$), and the rotation matrix(R) of the moving platform. This equation is derived from the inverse kinematics equation (1.1), and the forward kinematics equation can be written as [26],

$$v_i(\vec{l_i}, \vec{p_i}, R) = ||\vec{a_i} - R\vec{b_i} - \vec{p}||_2^2 - ||\vec{l_i}||_2^2$$

Define the sum of the absolute value of $v_i(\vec{l},\vec{p},R)$ for all cables be η, such that

$$\eta = \sum_{i=1}^{n} |v_i(\vec{l_i},\vec{p_i},R)|$$

and then the goal is to estimate $\vec{p}$ that minimizes η. The estimated position $\vec{p}$ has to satisfy (1.1), and the problem can be written as the optimization problem as follows,

$$\begin{aligned} \min_{\vec{p}} \quad & \eta \\ \text{subject to} \quad & 0 \leq \vec{p} \leq d \\ & \vec{a_k} - R\vec{b_k} - \vec{p} = \vec{l_k} \end{aligned} \tag{1.2}$$

where, k indicates the i-th cable. The position vector associated with cable lengths is determined by solving (1.2). At the static equilibrium condition, the sum of all forces due to cable tensions is equivalent to any external forces and moments applied to the platform. To express this equilibrium, the i-th cable's unit cable vector is defined as $\vec{u_i} = \vec{l_i}/|\vec{l_i}|$, and the tension vector of each cable as $\vec{\tau_i} = \vec{u_i}|\vec{\tau_i}|$. The following equations can be derived,

$$\sum_{i=1}^{n} \vec{u_i}|\vec{\tau_i}| = \vec{F}, \quad \sum_{i=1}^{n} R\vec{b_i} \times \vec{u_i}|\vec{\tau_i}| = \vec{M}$$

Defining the tension vector $\vec{\tau} = [|\vec{\tau_1}| \ \cdots \ |\vec{\tau_n}|]^T$, the static equation can be written as,

$$\vec{f} = J\vec{\tau}, \quad \text{with} \quad J = \begin{bmatrix} \vec{u}_1 & \cdots & \vec{u}_n \\ R\vec{b}_1 \times \vec{u}_1 & \cdots & R\vec{b}_n \times \vec{u}_n \end{bmatrix}_{m \times n} \tag{1.3}$$

where $\vec{f} = [\vec{F} \ \vec{M}]^T$ is the wrench vector, and m represents the number of degrees of freedom. Matrix $J \in \mathbf{R}^{m \times n}$ is called the Structure matrix, and the Singular Value Decomposition(SVD) analysis on J can determine the properties of tension solutions[5]. This will be discussed in detail in Chapter 3.1.1.

1.2 Dynamics

Defining the platform state vector $\vec{r} = [x\ y\ z\ \phi\ \theta\ \psi]^T$, and gravitational acceleration vector as $G = [0\ \ 0\ \ g\ \ 0\ \ 0\ \ 0]^T$ with the platform mass m and the gravitational acceleration g, the dynamic equation of the moving platform can be expressed as [27],

$$J\vec{\tau} = M\ddot{\vec{r}} + C\dot{\vec{r}} + MG, \tag{1.4}$$

where M, and C are the platform mass and Coriolis matrices defined by,

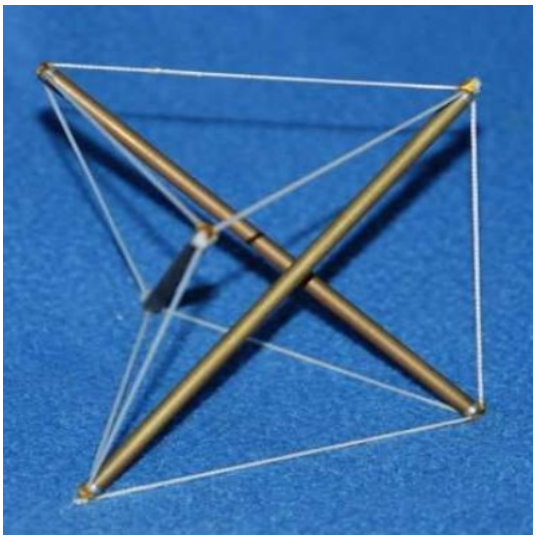
$$M = \begin{bmatrix} mI_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & I_{g3\times3} \end{bmatrix}, \quad C = \begin{bmatrix} 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & I_g + \omega \times I_g \end{bmatrix},$$

with the identity matrix I and the moment of inertia of a platform I_g. $\omega = [\omega_x\ \omega_y\ \omega_z]^T$ denotes the platform angular velocity vector. The term $J\vec{\tau}$ in (1.4) the force and moment vector induced by cable tensions. (1.4) This equation can be rewritten in matrix form with respect to the platform state vector $\vec{r}$,

$$\begin{bmatrix} \dot{\vec{r}} \\ \ddot{\vec{r}} \end{bmatrix} = \begin{bmatrix} 0 & I \\ 0 & -M^{-1}C \end{bmatrix} \begin{bmatrix} \vec{r} \\ \dot{\vec{r}} \end{bmatrix} + \begin{bmatrix} 0 \\ M^{-1}(J\vec{\tau} - MG) \end{bmatrix}. \tag{1.5}$$

Considering that cable tension $\vec{\tau}$ as the input of the dynamic model and extracting $\vec{\tau}$ from the matrix, (1.5) becomes,

$$\begin{bmatrix} \dot{\vec{r}} \\ \ddot{\vec{r}} \end{bmatrix} = \begin{bmatrix} 0 & I \\ 0 & -M^{-1}C \end{bmatrix} \begin{bmatrix} \vec{r} \\ \dot{\vec{r}} \end{bmatrix} + \begin{bmatrix} 0 \\ -G \end{bmatrix} + \begin{bmatrix} 0 \\ M^{-1}J \end{bmatrix} \vec{\tau}. \tag{1.6}$$



(a) Regular Minimal Tensegrity Prism with three bars and three cables.

(b) Kenneth Snelson's "Free Ride Home", 1974, an artwork leveraging tensegrity system

Figure 1.2. Examples of Tensegrity Systems

1.3 CDPRs as a Tensegrity System

Cable-Driven Parallel Robots (CDPRs) are composed of multiple cables and a moving platform, and these systems can be described as tensegrity systems. Figure 1.2 is an example of a tensegrity system structure. Tensegrity systems maintain stability as forces on cables and bars are balanced at the equilibrium. CDPRs maneuver a tensegrity system from its initial equilibrium to desired states. Leveraging the principles of tensegrity, CDPRs result in structures that are lightweight yet robust, particularly in handling high payloads. The use of cables serves to reduce both weight and inertia in CDPRs [10][28]. Analyzing the statics and dynamics of the tensegrity system becomes a valuable method to understand and characterize the distribution of forces within the bars and cables of the CDPR structure [29]. The study on the stabilization of multi-tethered balloons has utilized tensegrity analysis to optimize designs, ensuring that all cables remain taut even in the presence of disturbances such as winds [30]. This underscores the versatility of tensegrity principles in addressing stability and control challenges in different applications, including CDPRs.

In tensegrity systems, cables, and bars are typically referred to as members, and the points at which these members connect are defined as nodes. Members and nodes are denoted as vectors as follows,

$$M = [\vec{m}_1, \ldots, \vec{m}_p], \quad N = [\vec{n}_1, \ldots, \vec{n}_q],$$

where p is the number of members and q is the number of nodes, each member and node can be represented by a set of coordinates. The vector notation for each member is defined as the difference between two connected node vectors, denoted $\vec{m}_i = \vec{n}_j - \vec{n}_k$, where $i \in \{1, \ldots, p\}$ and $j,k \in \{1, \ldots, q\}$. The relationship between member and node vectors can be expressed by defining a connectivity matrix C, which is a $p \times q$ matrix, and can be written as follows,

$$M = NC^T.$$

The elements of a connectivity matrix define the connections between members and nodes, indicating which nodes are associated with each member. Each row of C corresponds to each $\vec{m}_i$ has one entry equal to 1, one entry equal to -1, and all other entries equal to zero. Define the sum of gravitational and inertial forces on the nodes as F and the internal forces applied by members as I. Let $\vec{d}_i$ be a unit member vector, and $\vec{x}_i$ be a tensile or compressive force in a member. Then, the unit member and forces can be written as,

$$D = [\vec{d}_1, \ldots, \vec{d}_p], \quad X = \text{diag}(x_1, \ldots, x_p).$$

The internal forces at each node can be defined in terms of X by using the connectivity matrix, such that,

$$I = DXC.$$

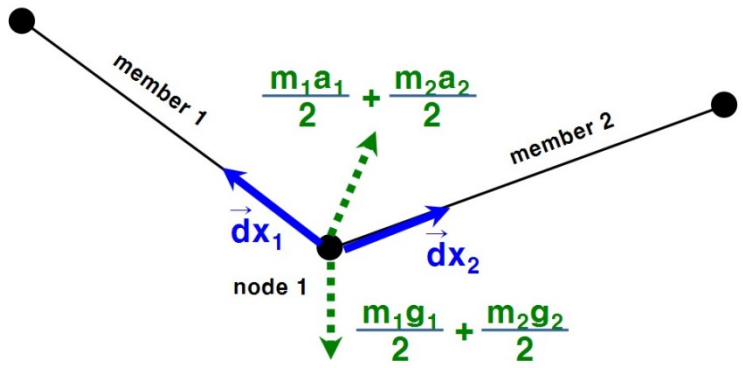


Figure 1.3. Free body diagram of tensegrity system's node and member. Two members are connected to node 1, and each member has two nodes. At node 1, the total force is the sum of the gravitation and inertial forces of two members. The forces of members are exerted uniformly at each node. The internal forces ($\vec{d}_1 x_1, \vec{d}_2 x_2$) are the sum of tensile or compression forces applied by members.

Figure 1.3 shows the example of all the forces exerted on a node. Newton's law can be leveraged to express the sum of forces at each node. The equation representing the equilibrium of forces at a node is given by,

$$F = [\vec{F}_1, \cdots, \vec{F}_p] = \left(\sum_{i=1}^{z} m_i(\vec{a}_i + \vec{g})/z \right)_p, \tag{1.7}$$

where m_i is the mass, and a_i is the acceleration vector of i-th member. g represents the gravitational acceleration vector, and z is the number of nodes i-th member has. (1.7) provides a concise expression of the forces and accelerations acting on nodes within a CDPR system when it is in a stationary state.

At equilibrium, forces in each node are balanced. The equation can be written as,

$$F = I \quad \Rightarrow \quad F = DXC. \tag{1.8}$$

Recalling that X is the diagonal matrix of tensile or compressive forces in each member, cable tensions can be calculated by solving equation (1.8) for X [11]. The right-hand side of (1.8) is a linear equation in terms of $[x_1, \cdots, x_p]$, so the equation can be rewritten in the form of a linear

matrix equation in terms of the vector $\vec{x} = [x_1, \cdots, x_p]^T$,

$$A\vec{x} = \vec{f} \tag{1.9}$$

where $\vec{f}$ is the magnitude of forces on each node, defined as $\vec{f} = [f_1, \cdots, f_p]^T$. Tension solutions can be obtained by solving equation (1.9). The general solution of the cable tension vector $\vec{x}$ can be written as,

$$\vec{x} = A^{\dagger}\vec{f} + V\vec{h} \tag{1.10}$$

where $A^{\dagger}$ is the pseudo inverse matrix of A, V is the null space of A, and h is the arbitrary vector. The solution can be optimized by choosing the arbitrary vector $\vec{h}$ if the null space V exists. The cable tension changes in cable tensions over time as the platform moves to the desired position. The objective is to maintain cable tensions within a specified boundary to prevent cables from becoming slack or damaged, the tension boundaries can be written as,

$$x_{min} \leq x \leq x_{max}$$

where x_{min} is the minimum tension limit, and x_{max} is the maximum tension limit.

Defining the lower and the upper bound tension margins at time t as, $\lambda_{t,min}, \lambda_{t,max}$ as the difference between the tension limits x_{min}, x_{max} and the minimum and the maximum tensions $(\min(x), \max(x))$. A system with larger tension margins has more capability of moving the platform to the tension limits. Tensions are required to be optimized around the mean value of the minimum and the maximum tension limits to maximize the tension margins. Maximizing tension margins is equivalent to minimizing the standard deviation of tensions relative to x_{ave}. Leveraging the equation of standard deviation, the optimization problem can be written as a

convex problem as follows,

$$\begin{aligned} \underset{\vec{x},\vec{h}}{\text{argmin}} \quad & \sqrt{\frac{\sum_{i=1}^{m}(x_i - x_{ave})^2}{b}} \\ \text{subject to} \quad & x_{min} \leq x \leq x_{max} \\ & x_{ave} = (x_{min} + x_{max})/2 \\ & \vec{x} = A^{\dagger}\vec{f} + V\vec{h} \end{aligned} \tag{1.11}$$

Solving (1.11) provides the optimal tension solution bounded in the desired tension boundaries.

1.4 Acknowledgements

Support for this work was provided by ROK Navy Sponsored Education Program.

Chapter 1, in part,is a reprint of the material as it appears in International Conference on Robotics Systems and Automation Engineering(RSAE) 2021, J. Jang and T. Bewley, “Tension optimization of the 6-DOF cable-driven boat motion simulator”. The dissertation author was the primary investigator and author of this paper.

Chapter 2

Control and Tension Optimization

Chapter 2 investigate the control aspect of the simulator. Although (1.11) provides optimal tension solutions derived from the static equation when a platform is in a stationary pose, CDPRs involve dynamic motion with various accelerations and poses. Therefore, the controller for CDPRs is required to handle both dynamic motion control and tension optimization. To address these objectives, we propose a Control Lyapunov Function-based Quadratic Program (CLF-QP) controller design method. This chapter will discuss the details of Control Lyapunov Function theories and their application in CDPR controls.

2.1 Control Lyapunov Function

Consider the following dynamic system,

$$\dot{x} = f(x) + g(x)u, \tag{2.1}$$

where x is the state vector, $f(x)$ and $g(x)$ are continuous functions of x, and u is the feedback control input that stabilizes the system. Assuming $V(x)$ is a smooth, positive definite function, define $L_f V(x)$, and $L_g V(x)$ as follows,

$$L_f V(x) = \frac{\delta V}{\delta x} f(x), \quad L_g V(x) = \frac{\delta V}{\delta x} g(x). \tag{2.2}$$

Then, a function $V(x)$ that satisfies the following inequality,

$$\dot{V}(x) = L_f V(x) + L_g V(x) u < 0, \quad \forall x \neq 0, \tag{2.3}$$

is called a Control Lyapunov Function (CLF)[31]. It provides a sufficient condition for the stabilizability of systems. The existence of CLFs indicates that there exists feedback control u that stabilizes the system[32], and the inequality (2.3) implies that the system is asymptotically stable. However, this formulation does not provide information about the convergence rate to the equilibrium. To address this, one can construct the upper bound of $V(x)$ and impose a condition on the minimum rate of decrease in $V(x)$. (2.3) can be rewritten with an additional term $\lambda V(x)$ [33],

$$\dot{V}(x) = L_f V(x) + L_g V(x) u + \lambda V(x) < 0, \quad \forall x \neq 0, \tag{2.4}$$

where λ represents the decay rates of $V(x)$, influencing the convergence characteristics of the system.

2.2 Application of CLF to CDPR dynamics

Let $\vec{x}$ be the state of a moving platform such that $x = [\vec{r} \;\; \dot{\vec{r}}]$ with the state vector $\vec{r} = [x \; y \; z \; \phi \; \theta \; \psi]^T$ as defined in 1.2. Defining the control input $u = [\vec{\tau}]$ as a cable tension vector. By the dynamic equation (1.6) and (2.1), functions $f(x)$ and $g(x)$ can be defined as,

$$f(x) = \begin{bmatrix} 0 & I \\ 0 & -M^{-1}C \end{bmatrix} \begin{bmatrix} \vec{r} \\ \dot{\vec{r}} \end{bmatrix} + \begin{bmatrix} 0 \\ -G \end{bmatrix},$$
$$g(x) = \begin{bmatrix} 0 \\ M^{-1}J \end{bmatrix}. \tag{2.5}$$

The chosen CLF is $V(x) = ||x - x_d||_2^2$, where $x_d = [\vec{r}_d \;\; \dot{\vec{r}}_d]$ is the desired platform state. The gradient of $V(x)$ is given by,

$$\frac{\delta V}{\delta x} = 2(x - x_d),$$

and by (2.2), $L_f V(x)$, and $L_g V(x)$ can be expressed as,

$$L_f V(x) = 2(x - x_d)^T \Big(\begin{bmatrix} 0 & I \\ 0 & -M^{-1}C \end{bmatrix} \begin{bmatrix} \vec{r} \\ \dot{\vec{r}} \end{bmatrix} + \begin{bmatrix} 0 \\ -G \end{bmatrix} \Big),$$

$$L_g V(x) = 2(x - x_d)^T \begin{bmatrix} 0 \\ M^{-1}J \end{bmatrix}.$$

2.3 CLF-based Quadratic Program

As discussed before, CDPR controllers must address the challenge of controlling platform poses and maintaining cable tensions as the platform moves to desired positions. The Quadratic Program (QP) is employed to achieve these objectives, preserving performance expectations set by the CLF and adhering to user-defined constraints on the inputs[16][24]. The CLF-QP is formulated as a convex optimization problem, with the standard form represented as[25]:

$$\begin{aligned} \underset{\vec{u}}{\text{argmin}} \quad & (\vec{u} - \vec{u}_d)^T (\vec{u} - \vec{u}_d) \\ \text{subject to} \quad & L_f V(x) + L_g V(x) + \lambda V(x) \leq 0, \end{aligned} \tag{2.6}$$

where $\vec{u}_d$ is the reference input. The CLF-QP seeks to minimize the deviation of the input from the reference while satisfying the CLF conditions and any additional constraints.

As discussed in 1.3, the tension margin represents the difference between current tensions and the maximum or minimum tension limits. Optimizing tensions near the mean value of the

maximum and minimum tension limits is crucial to maximizing tension margins[14]. To achieve maximizing the tension margins, the reference input $\vec{u}_d$ can be set as the mean value of tension limits. However, the optimal cable tensions must not only be within tension limits but also satisfy the dynamic equation. Since these hard constraints severely limit solution sets, there is a need to relax the bound on the CLF to minimize conflicts with constraints. By introducing additional constraints and relaxation term, the original CLF-QP problem (2.6) is modified as follows,

$$\begin{aligned} \underset{\vec{u}}{\text{argmin}} \quad & (\vec{u}-\vec{u}_d)^T(\vec{u}-\vec{u}_d)+p\delta^2 \\ \text{subject to} \quad & L_fV(x)+L_gV(x)+\lambda V(x)\leq\delta \\ & \vec{u}_{min}\leq\vec{u}\leq\vec{u}_{max} \\ & J\vec{u}=M\ddot{\vec{r}}_d+C\dot{\vec{r}}_d+MG, \end{aligned} \tag{2.7}$$

where δ is a positive constant that relaxes CLF constraints, and p is a positive number representing the penalty associated with δ. u_{min} and u_{max} are the minimum and maximum tension limits set by the user. The optimal control input $u=[\vec{\tau}]$ can be directly determined by solving (2.7).

2.4 Acknowledgements

Support for this work was provided by Republic of Korea(ROK) Navy Sponsored Education Program.

Chapter 2, in full, is a reprint of the material as it appears in IEEE International Conference on Robotics and Automation Engineering(ICRAE) 2022, J. Jang, M. Zhao, and T. Bewley, "A Control Lyapunov Function-Based Quadratic Program for the Cable-Driven Boat Motion Simulator". The dissertation author was the primary investigator and author of this paper.

Chapter 3

Workspace Analysis for Parameter Optimization

This chapter delves into the workspace analysis of a cable-driven boat motion simulator for parameter optimization. The objective is to optimize design parameters that is optimal to simulate the desired motions. The simulator is designed to replicate the 6-DOF motion of the USS Oliver Hazard Perry-class frigate. The frigate's motion characteristics include a maximum roll angle of 30 degrees and a sway magnitude of 5 centimeters on a 1/100 scale [34]. The simulator, equipped with eight cables and a platform, necessitates an ample range for 6-DOF boat motion. The primary goal of this chapter is to ascertain the workspace of the cable-driven boat motion simulator, ensuring its capability to replicate the desired boat motions by providing an appropriate range of motion. Three key design parameters were taken into account: platform height, width, and cable attachment points on the platform. The simulator's workspace is influenced by these parameters, and simulation tests were conducted to observe the impact on the workspace. Three methods were explored to determine the simulator's workspace. Firstly, as the simulator comprises eight cables and operates at 6-DOF, it falls into the category of an underdetermined system. The tension solutions can be optimized at the equilibrium state within this system[17]. Singular Value Decomposition (SVD) was applied to the static equilibrium equation to analyze the tension solutions. The rank of the Structure matrix in the static equation was examined to determine whether the underdetermined CDPR has infinite

or no solutions[30]. Additionally, considering the CDPR with multiple cables, we addressed cable-to-cable and cable-to-platform interference, as these factors influence the determination of the workspace. For cable-to-cable interference, the shortest distance between cables was calculated using the magnitude of the perpendicular vector formed by two cable vectors obtained through inverse kinematics[35]. Cable-to-platform interference was detected when a cable intersected or passed through the side, upper, or bottom planes of the platform. Cable attachment vectors were used to define the planes of the platform. By comparing the angle between a cable vector and the neighboring platform plane, it was determined whether interference occurred. An angle of zero indicated that the cable was intersecting or passing through the platform, leading to interference[9]. Based on these methods, an algorithm was developed to determine the workspace of the cable-driven boat motion simulator. The algorithm integrated static analysis and interference detection methods. A simulation model of the boat motion simulator was employed to analyze the workspace and apply the algorithm, offering insights into the achievable workspace and potential interference within the system.

3.1 Workspace Analysis Methods

3.1.1 Static Analysis

Revisiting the static equilibrium equation represented as (1.3), J is a $m \times n$ matrix where m is DOF, and n is the number of cables connected to a platform. Conducting a Singular Value Decomposition (SVD) analysis on the Structure matrix J yields valuable insights into discerning feasible solutions. The SVD is applicable to any matrices, leading to matrix J in (1.3) having an SVD expressed as $J = U\Sigma V^T$, where $U_{m\times m}$ and $V_{n\times n}$ are unitary matrices, and $\Sigma_{m\times n}$ is a diagonal matrix containing the singular values σ_i of J on its main diagonal[36]. An SVD of the Structure matrix J, can be rewritten as follows with r, the rank of J

The Structure matrix J can be reformulated using the rank r of J as follows[37],

$$\begin{aligned} J_{m\times n} &= U\Sigma V^T \\ &= \begin{bmatrix} \underline{U}_{m\times r} & \overline{U}_{m\times(m-r)} \end{bmatrix} \begin{bmatrix} \underline{\Sigma}_{r\times r} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \underline{V}_{n\times r} & \overline{V}_{n\times(n-r)} \end{bmatrix}, \end{aligned} \tag{3.1}$$

where,

- $\underline{U}$ represents the column space, and $\overline{U}$ is the left null space of J,
- $\underline{V}$ represents the row space, and $\overline{V}$ is the null space of J.

Utilizing the null space of J, denoted as $J\overline{V} = 0$, the tension solution is represented as [38],

$$\vec{\tau} = J^{\dagger}\vec{f} + \overline{V}\vec{h}, \tag{3.2}$$

where $J^{\dagger}$ is the pseudo inverse matrix of J, and $\vec{h}$ is an arbitrary vector. The choice of $\vec{h}$ in (3.2) results in optimal tension solutions at a specific platform pose. By performing an SVD on the matrix J, (1.3) can be categorized as,

- An underdetermined with ∞ tension solutions if ($n > r = m$),
- Both potentially inconsistent and underdetermined with either ∞ or 0 solutions if ($n > r$, $m > r$) depending on $\vec{f}$,
- A determined with only 1 unique tension solution if ($m = n = r$),
- An overdetermined with either 0 or 1 tension solution if ($m > r = n$), depending on $\vec{f}$.

As the boat motion simulator comprises eight cables and exhibits 6-DOF, it can be categorized as an underdetermined system. In the case where J is of full row rank, there are infinite number of solutions, allowing for optimization of tension solutions. The solution vector $\vec{\tau}$ needs to consist

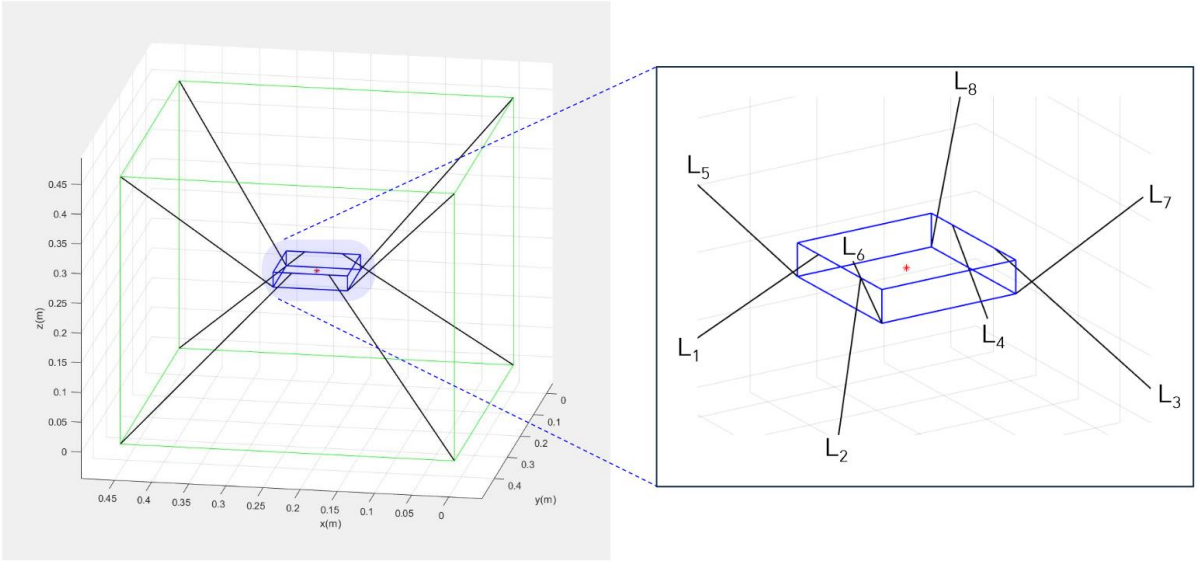
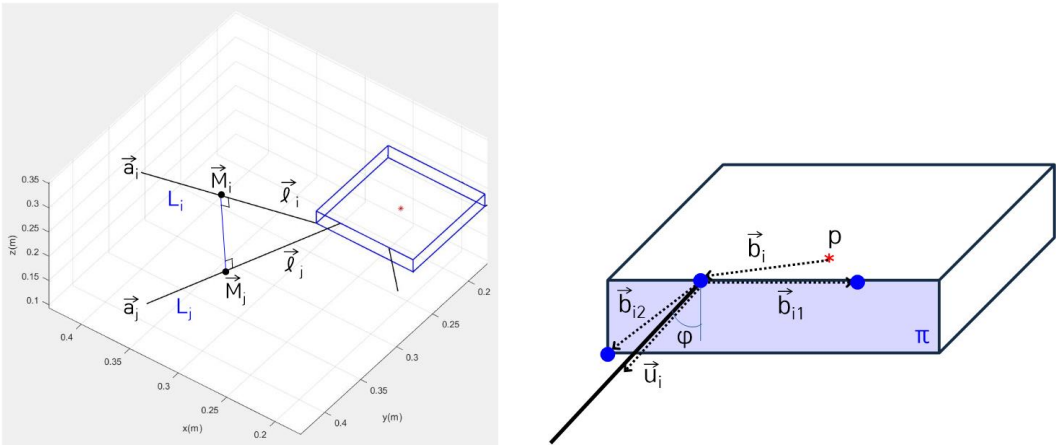


Figure 3.1. Cable-driven Boat Motion Simulator simulation model. The green lines represent an outer structure, and eight cables(black) are connected to a moving platform(blue). The figure on the right-hand side provides a zoomed-in perspective of the cables ($L_1 \sim L_8$) and the platform.

of positive tensions to exert forces on the platform, given that cables can only pull an object and not push it. Additionally, tensions should be minimized to prevent structural or cable damage from excessively high tensions. The optimal tension solution $\vec{\tau}$ can be obtained by determining the arbitrary vector $\vec{h}$ in (3.2) through the following optimization problem [39],

$$\begin{aligned} \arg\min_{\vec{h}} \quad & \vec{\tau} \\ \text{subject to} \quad & \vec{\tau} = J^{\dagger}\vec{f} + \overline{V}\vec{h} \\ & 0 \leq \vec{\tau} \end{aligned} \tag{3.3}$$

Once the vector $\vec{h}$ is determined by solving the optimization problem, it indicates that the platform is located within the feasible workspace, with the tension solution satisfying the conditions in (3.3).



(a) Cable-to-Cable interference. **(b)** Cable-to-Platform interference.

Figure 3.2. Interference Detection

3.1.2 Cable-to-Cable Interference

Figure 3.1 illustrates that the upper and lower cables are interconnected to the platform or expanding the rotational workspace, instead of direct connections to the platform's edges. As discussed in 3.1.1, static analysis aids in determining tension solutions within the feasible workspace for a given platform pose. However, the presence of crossing cables introduces interference within the feasible workspace, imposing constraints on the platform's motion. It is essential to address these interferences to ensure unrestricted movement. Cable-to-cable interference occurs when two cables intersect during the platform's motion. Detecting such interference involves calculating the distance between two cables. If this distance becomes smaller than the diameter of a cable, it indicates interference. In Figure 3.2a, $\vec{l}_i$ and $\vec{l}_j$ represent vectors corresponding to cables L_i and L_j, while $\vec{a}_i$ and $\vec{a}_j$ denote pulley position vectors on the outer frame. The vector $\vec{M}_{ij} = (\vec{M}_i - \vec{M}_j)$ is perpendicular to both cable vectors, and $\vec{M}_i$ and $\vec{M}_j$ can be defined as follows[35],

$$\vec{M}_i = \vec{a}_i + \lambda_i \vec{u}_i,$$

$$\vec{M}_j = \vec{a}_j + \lambda_j \vec{u}_j,$$

where the unit cable vectors are defined as $\vec{u}_i = \vec{l}_i/|\vec{l}_i|$ and $\vec{u}_j = \vec{l}_j/|\vec{l}_j|$. The parameters λ_i and λ_j represent the magnitudes of vectors. The cross product $(\vec{u}_i \times \vec{u}_j)$ is a direction vector of $\vec{M}_{ij}$. The calculation of the shortest distance d_{ij} between two cables involves determining the length of the orthogonal projection of $(\vec{a}_i - \vec{a}_j)$ onto $\vec{M}_{ij}$. The formula for this calculation is given by[40],

$$d_{ij} = \Big| \frac{(\vec{a}_i - \vec{a}_j) \cdot (\vec{u}_i \times \vec{u}_j)}{|\vec{u}_i \times \vec{u}_j|} \Big|. \tag{3.4}$$

If d_{ij} is smaller than the cable diameter ($d_{ij} \leq \varepsilon$), it indicates the interference between two cables, L_i and L_j.

3.1.3 Cable-to-Platform Interference

Figure 3.2b depicts a platform and a cable connected to the attachment point. A vector $\vec{b}_i$ represents the attachment point vector of the i-th cable in the body frame on the platform center p. $\vec{b}_{i1}$ and $\vec{b}_{i2}$ are the vectors from the attachment point of i-th cable to the neighboring cable attachment points. A plane π formed by two vectors, and the ϕ iss defined as the angle between π and the unit cable vector $\vec{u}_i$. To detect the interference, the perpendicular vector of plane π needs to be defined using the given vectors $\vec{b}_{i1}$ and $\vec{b}_{i2}$. The perpendicular vector $\vec{q}$ can be calculated through the vector cross-product,

$$\vec{q}_i = \vec{b}_{i1} \times \vec{b}_{i2}.$$

Note that the angle between $\vec{q}_i$ and $\vec{u}_i$ is $\frac{\pi}{2} - \phi$ and this angle is formulated using the dot product,

$$\cos(\frac{\pi}{2} - \phi) = \frac{\vec{q}_i \cdot \vec{u}_i}{|\vec{q}_i||\vec{u}_i|},$$
$$\phi = \sin^{-1}\big(\frac{\vec{q}_i \cdot \vec{u}_i}{|\vec{q}_i||\vec{u}_i|}\big).$$

Cable-to-platform interference occurs when the angle ϕ is less than zero.

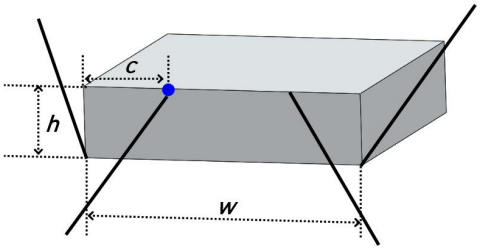


Figure 3.3. Design Parameters: The platform height(h), width(w), and the cable attachment point(c).

3.2 Simulation Setup

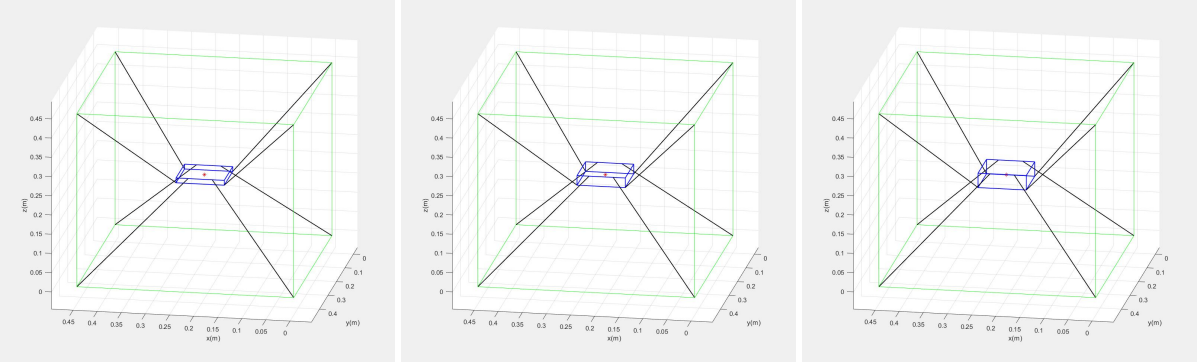
3.2.1 Simulation Model

The simulation model for the cable-driven boat motion simulator is depicted in Figure 3.1. The outer structure of the model is a cube, and a moving platform is connected to eight cables. The upper and lower cables are cross-connected, with the lower cables attached inward from the top edge of the platform. The simulation test involved moving the platform to every position within the structure, with a resolution of 1cm for translation in the x, y, and z axes. The test considered eight cases of platform rotation based on the desired motion, USS Oliver Hazard class Frigate on the sea state 6[1]. These cases include roll angles from 0 to 30 degrees, pitch angles from 0 to 5 degrees, and yaw angles from 0 to 2 degrees. Combinations of these rotations are detailed in TABLE 3.1.

Table 3.1. Rotation Motion Test Sets

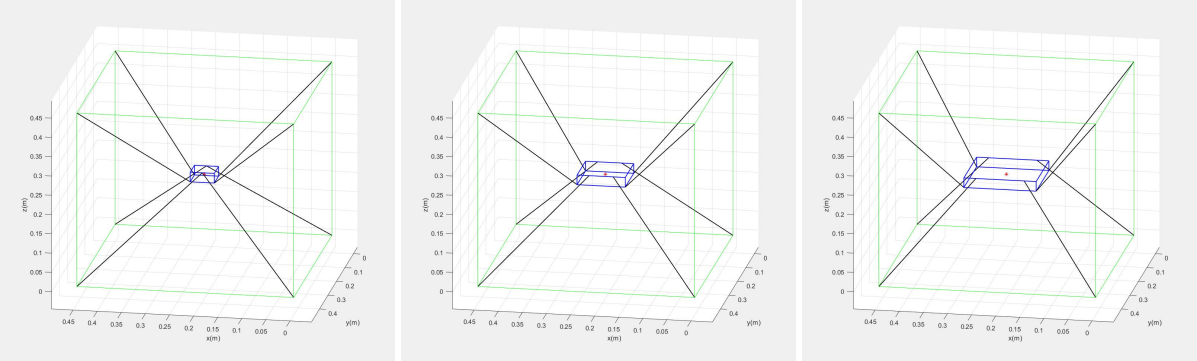
Case	Roll, Pitch, Yaw	Case	Roll, Pitch, Yaw
1	0, 0, 0	5	30, 0, 0
2	0, 0, 2	6	30, 0, 2
3	0, 5, 0	7	30, 5, 0
4	0, 5, 2	8	30, 5, 2

[1]Douglas sea scale 6: Very rough sea state with $4 \sim 6$ meters wave heights.



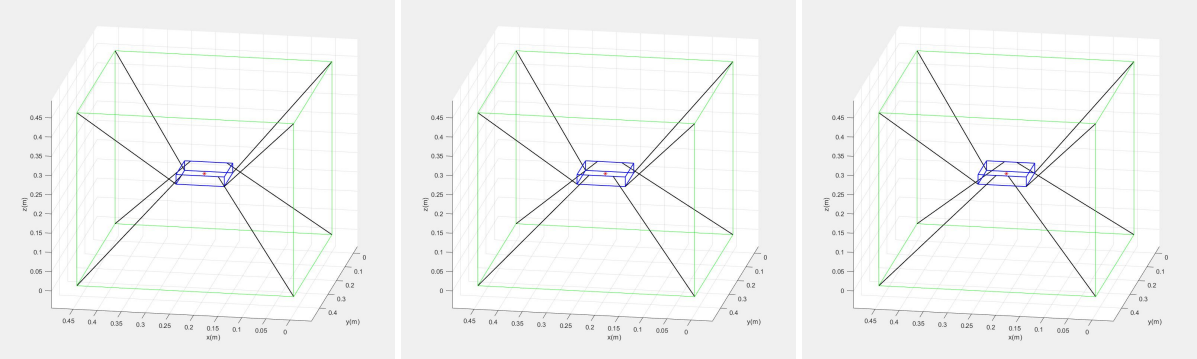
(a) H1: h = 1.25cm **(b)** D0: h = 2.50cm **(c)** H2: h = 3.75cm

Figure 3.4. Platform Height with w = 10.00cm, c = 2.50cm



(a) W1: w = 5.00cm **(b)** D0: w = 10.00cm **(c)** W2: w = 15.00cm

Figure 3.5. Platform Width with h = 2.50cm, c = 2.50cm



(a) C1: c = 1.25cm **(b)** D0: c = 2.50cm **(c)** C2: c = 3.75cm

Figure 3.6. Cable Attachment Point with h = 2.50cm, w = 10.00cm

3.2.2 Design Parameters

During the workspace analysis, three design parameters were investigated, as illustrated in Figure 3.3. The impact of variations in these parameters was evaluated by adjusting them by 50 percent, either decreasing or increasing, from the baseline design denoted as 'D0.'

Platform Height

The first parameter is the platform height, which signifies the distance between the bottom and top planes of the platform. Three distinct heights were examined to explore the correlation between the platform height and the workspace, as illustrated in Figure 3.4. The baseline design, denoted as 'D0,' features a platform height of 2.50cm, while 'H1' exhibits a decreased height of 1.25cm, and 'H2' has an increased height of 3.75cm.

Platform Width

The second parameter is the platform width, representing the gap between the edges of the top or bottom plane. Diverse width dimensions were tested to assess the impact of platform width on the workspace, as depicted in Figure 3.5. The baseline design, labeled 'D0,' maintains a platform width of 10.00cm, while 'H1' features a reduced width of 5.00cm, and 'H2' showcases an increased width of 15.00cm.

Cable Attachment Point

The third parameter related to the cable attachment point, involving the attachment of lower cables to the edges of the upper platform plane, positioned towards the middle of these edges to enhance rotational motion. Represented as b in Figure 3.6, the cable attachment point indicates the distance from the nodes to the attachment points. Various values for the attachment point were examined to gauge their impact on the workspace. As depicted in Figure 3.6, the baseline design 'D0' features a cable attachment point of 2.50cm, 'H1' adopts an attachment point of 1.25cm, and 'H2' is characterized by an attachment point of 3.75cm.

By modifying these design parameters and assessing their effects on the workspace, valuable insights can be gained regarding how height, width, and cable attachment points impact the range of achievable motion in the cable-driven boat motion simulator. This analysis facilitates the optimization of design parameters to achieve an ideal workspace that accurately replicates the intended boat motions.

3.3 Simulation Algorithm

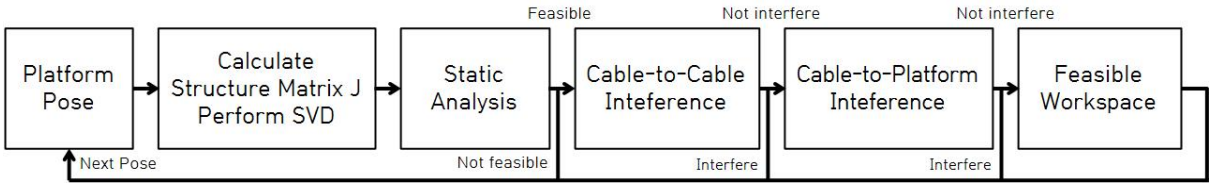


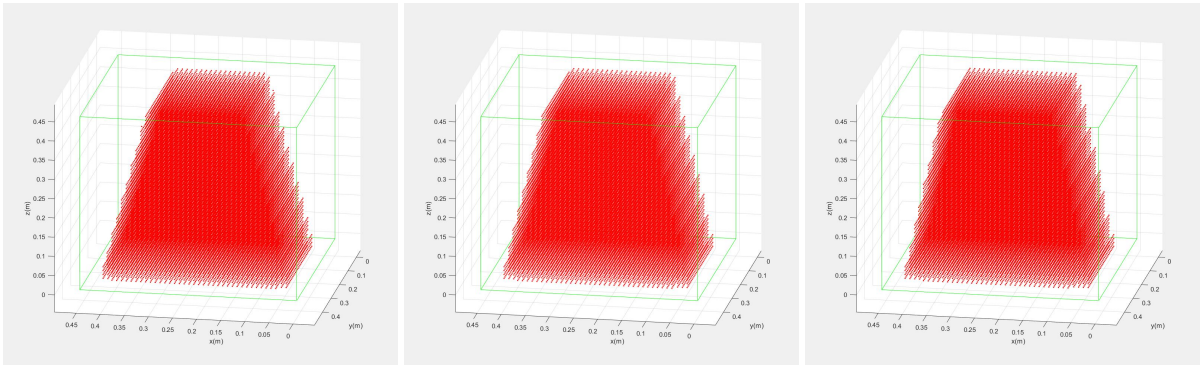
Figure 3.7. Simulation test algorithm.

The simulation test was executed following the algorithm outlined in Figure 3.7. The initial platform position was set at the origin. Utilizing equations (1.3) and (3.1), the corresponding structure matrix, J, was computed, and Singular Value Decomposition (SVD) was employed to determine the null space of J. The static analysis was conducted to explore the existence of $\vec{h}$ in (3.3). If a feasible tension solution was identified, indicating the existence of $\vec{h}$, the algorithm proceeded to investigate cable-to-cable and cable-to-platform interference using the methods detailed in 3.1.2 and 3.1.3. Finally, the feasible, interference-free platform pose was recorded, and the platform advanced to the next pose in the simulation.

3.4 Simulation Results

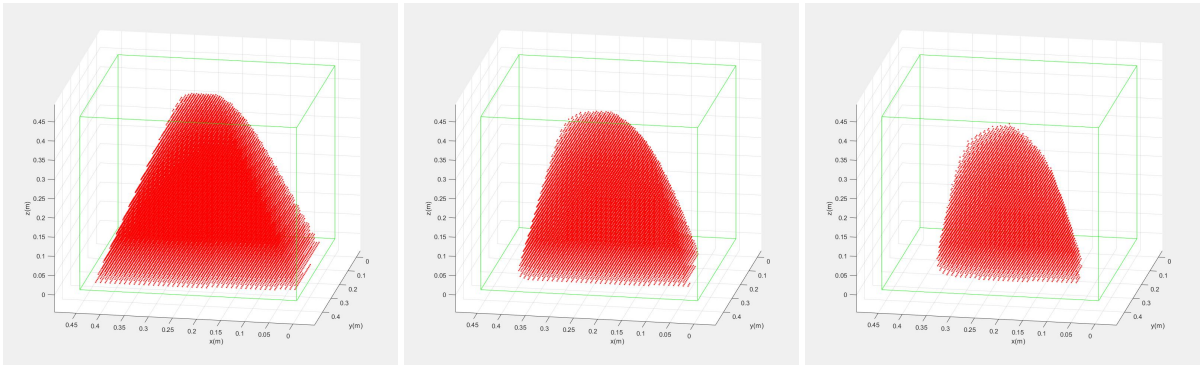
The workspace analysis includes a comparison of the workspace volumes for each design. The results are summarized in Table 3.2, showing the percentage of feasible workspace occupied by a platform center position out of the total number of positions within the structure. The results indicate that the workspace is most extensive for each design when the platform remains

non-rotational, and it gradually decreases as rotational motion is introduced. Additionally, the influence of rotations on each design's workspace is evident from the distinct variations observed.



(a) H1 (49.2%) **(b)** D0 (50.7%) **(c)** H2 (51.6%)

Figure 3.8. Case 1 Feasible Workspace (Platform Height)

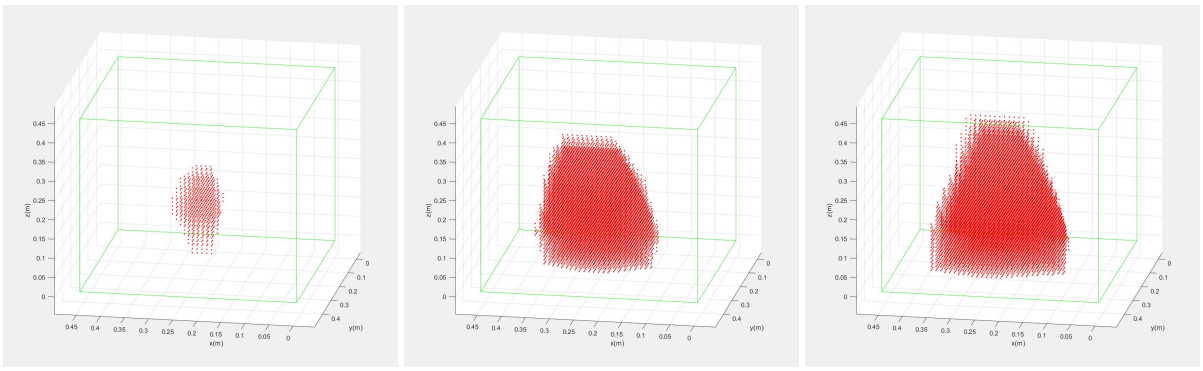


(a) W1 (43.6%) **(b)** D0 (30.3%) **(c)** W2 (18.3%)

Figure 3.9. Case 2 Feasible Workspace (Platform Width)

3.4.1 Platform Height

The variations in platform height (H1→D0→H2) have a relatively minor impact on the feasible workspace volume compared to the influences of other parameters. Figure 3.8 illustrates the feasible workspace of Case 1 for designs with different heights. The difference between the smallest and largest workspace ranges from a minimum of 1.0% to a maximum of 3.7%. An increase in platform height results in expanded workspace for motions without a roll (Case 1∼4).



(a) C1 (0.8%) **(b)** D0 (13.0%) **(c)** C2 (17.2%)

Figure 3.10. Case 6 Feasible Workspace (Cable Attachment Point)

Table 3.2. Feasible Workspace Volume (%)

Case	D0	H1	H2	W1	W2	C1	C2
1 (0,0,0)	50.7	49.2	51.6	54.6	40.3	50.4	47.1
2 (0,0,2)	30.3	29.7	30.8	43.6	18.3	17.4	34.8
3 (0,5,0)	49.3	47.3	50.8	53.4	39.1	48.9	45.4
4 (0,5,2)	31.5	30.9	31.9	44.7	19.1	18.2	35.8
5 (30,0,0)	16.5	16.9	13.6	26.2	12.9	8.7	18.7
6 (30,0,2)	13.0	12.9	10.8	25.7	6.6	0.8	17.2
7 (30,5,0)	14.7	14.8	11.6	25.6	9.3	1.9	17.8
8 (30,5,2)	15.1	15.1	11.4	25.1	9.7	3.8	18.1

However, a taller platform negatively affects motions involving a roll (Case 5$\sim$8) and increases the possibility of cable-to-platform and cable-to-cable interferences.

3.4.2 Platform Width

Changes in platform width (W1$\rightarrow$D0$\rightarrow$W2) have the most substantial impact on the feasible workspace. Alterations in platform width result in a workspace modification ranging from a minimum of 13.3% to a maximum of 25.6%. A smaller platform width is associated with reduced interference and enhanced tolerance for rotational movements, particularly in yaw (Case 2, 4, 6, 8). Figure 3.9 visually represents the feasible workspace of Case 2 for each design, illustrating a reduction in workspace volume attributed to yaw motion.

3.4.3 Cable Attachment Point

As cable attachment points are altered (C1→D0→C2), the feasible workspace undergoes variations ranging from a minimum of 3.3% to a maximum of 17.6%. If the cable attachment point is smaller and positioned close to the corners, the workspace contracts, particularly in the presence of rotations, especially roll. This outcome is due to the increased proximity of two cables at the corners, elevating the possibility of cable-to-cable interference. Figure 3.10 illustrates this effect, demonstrating how a larger cable attachment point significantly enlarges the workspace in case 6.

3.5 Analysis

This chapter delves into the analysis of workspace optimization for a Cable-driven Boat Motion Simulator. The study investigates three parameters—platform height, width, and cable attachment point—and examines their impact on the feasible workspace. The results reveal that increasing platform height leads to a smaller workspace during platform rolls but results in a larger workspace without rolls. Platform width significantly influences the workspace, showing that smaller platforms consistently offer a larger feasible workspace in all cases. Additionally, a smaller cable attachment point located close to the platform's corners reduces the workspace due to a higher likelihood of cable-to-cable interference. Conversely, a larger cable attachment point generally yields a larger workspace due to increased cable distance. The simulation test findings suggest that smaller platform widths and larger cable attachment points contribute to achieving a more extensive workspace. However, practical constraints, such as ensuring sufficient space for electronics and sensors on the platform, and allowing for landing tests in specific applications (e.g., aerial vehicle landings on a moving platform), must be taken into consideration.

In the future, efforts will concentrate on refining and determining optimal parameters to establish an ideal workspace that efficiently replicates a boat's motions. This will involve conducting additional experiments built upon the insights gained from the current study. Once the

optimal parameters are identified, the next step will be the construction of a hardware prototype model. Furthermore, the investigation will continue with additional simulation tests aimed at precisely formulating the relationship between parameters and workspaces. This iterative process aims to enhance the accuracy and effectiveness of the Cable-driven Boat Motion Simulator.

3.6 Acknowledgements

Support for this work was provided by ROK Navy Sponsored Education Program.

Chapter 3, in full, is a reprint of the material as it appears in IEEE International Conference on Control and Robotics(ICCR) 2023, J. Jang, and T. Bewley, “Workspace Analysis for Cable-Driven Parallel Robot based Boat Motion Simulator”. The dissertation author was the primary investigator and author of this paper.

Chapter 4

CDPR Simulator Experimental Validation

This chapter mainly discusses the experimental validation of the proposed CLF-QP control algorithm outlined in 2. To assess the efficacy of the control algorithm, we established both simulation and hardware models for the CDPR-based boat motion simulator. The chapter is structured as follows: Section 4.1 details the simulation model test, while Section 4.2 delves into the hardware test, incorporating eight cables and a moving platform model.

4.1 Simulation Test

In this chapter, we scrutinize the analysis of simulation results, primarily aimed at validating the effectiveness of the CLF-QP method. The simulation framework utilized in our investigation features a 6-DOF boat motion simulator equipped with eight cables and a platform. To assess the CLF-QP method, we exposed the simulator to two distinct motion scenarios: one marked by periodic motion and the other generated through the application of a mathematical ship model.

4.1.1 Simulation Model and Motion Scenario

The simulation model configuration is illustrated in Figure 4.1. Configurations of a periodic motion simulation test over time are depicted in Figure 4.2. The model's dimensions are built based on the design 'D0', equivalently matched to the hardware model. The outer structure's edge measures 0.45m, and the platform is a flat cuboid plate with dimensions of 0.10m. The

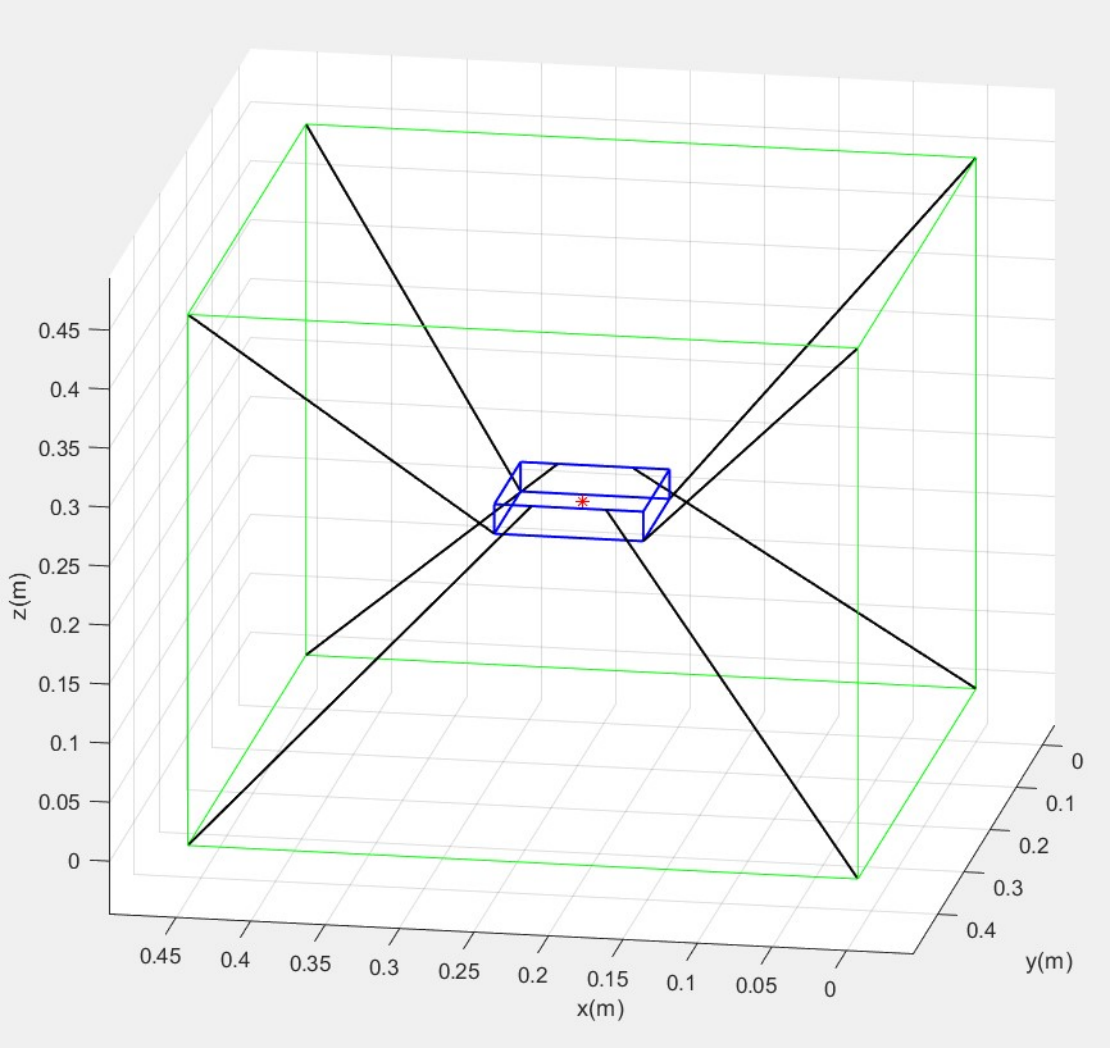
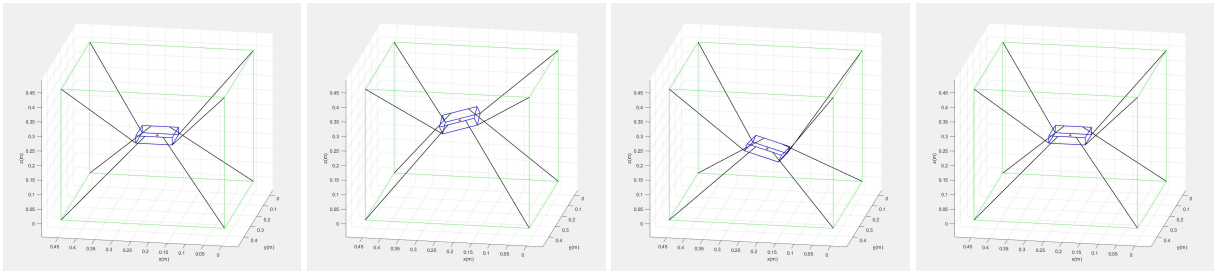
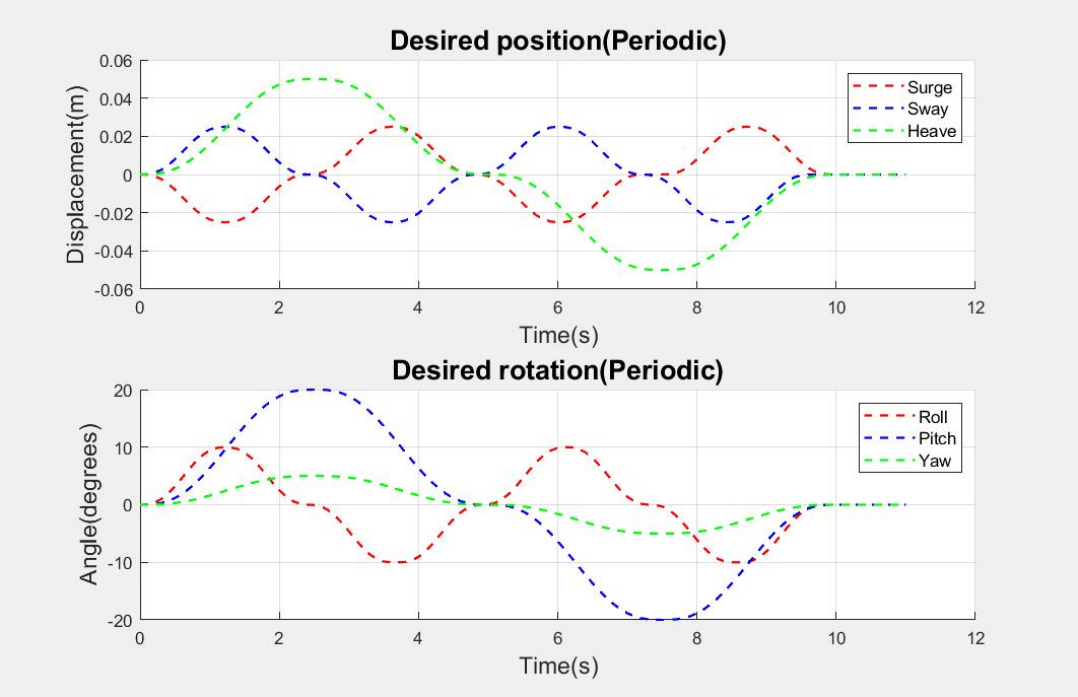


Figure 4.1. The simulation model of the cable-driven boat motion simulator. The Green lines are the outer structure, the black lines are cables, the blue lines are the platform, and the red dot is the mass center of the platform.

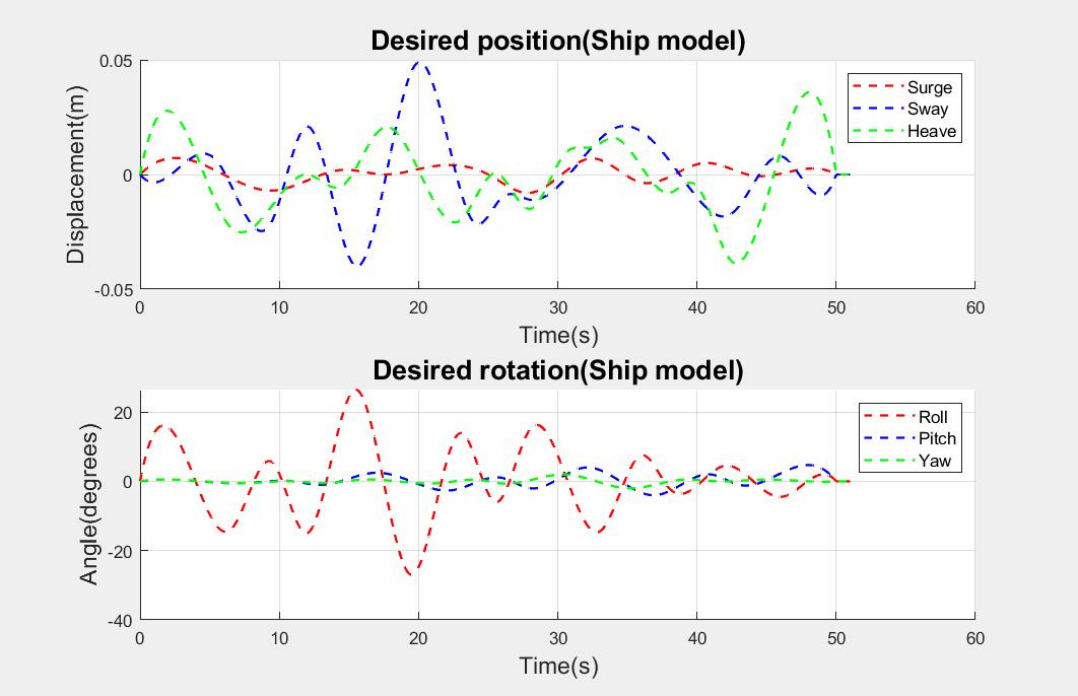


(a) t=0s **(b)** t=3s **(c)** t=7s **(d)** t=11s

Figure 4.2. The platform configurations over time(Periodic).



(a) Periodic motion



(b) Ship model motion

Figure 4.3. The desired motions of a platform.

Table 4.1. Desired motion of a platform: Periodic

Motion	Range	Period(seconds)
Surge	-0.025 – 0.025 m	5
Sway	-0.025 – 0.025 m	5
Heave	-0.050 – 0.059 m	10
Roll	-10°– 10°	5
Pitch	-20°– 20°	10
Yaw	-5°– 5°	10

platform's height is 2.5cm, and its mass is 0.858kg. The lower cables are connected to the inward point on the top, while the upper cables are connected to the bottom edges of the platform. Cables are cross-attached to the platform to enhance rotational motions. Cable mass is assumed to be negligible, and structures are not deformed by tensions. The cable material is Kevlar, which has a high Young's modulus ($76 \times 10^9 N/m^2$). Thus, cables are treated as non-stretchable. The maximum and minimum tension limits are set as 2N and 30N, respectively. The platform moves in 6-DOF motions for 10 seconds (periodic) and 50 seconds (ship model). After the motion, both cases have a 1-second stationary state at the origin. In this simulation scenario, at time $t = 5s$, a 1kg mass drone lands on the platform. Hence, there is additional mass on the platform after 5 seconds. The simulation time-step size is 0.1 seconds, and the CVX solver, a MATLAB package for specifying and solving optimization programs, is used[41][42].

4.1.2 Results

Tracking Errors

The desired platform motions are presented in Figure 4.3. Initially, a periodic motion was applied to the simulation, with detailed motion characteristics shown in 4.3a and summarized in Table 4.1. Figure 4.4a illustrates the translational and rotational tracking errors, revealing position errors ranging from -5.60×10^{-3} to 5.60×10^{-3} meters and rotation errors spanning from -2.29 to 2.23 degrees.

Next, Figure 4.3b illustrates the desired motion generated by the mathematical ship

Table 4.2. Motion tracking RMSE, NRMSE

Motion	RMSE(NRMSE): Periodic	RMSE(NRMSE): Ship model
Surge	0.0034m (6.73%)	0.0003m (2.22%)
Sway	0.0034m (6.73%)	0.0018m (1.97%)
Heave	0.0034m (3.39%)	0.0016m (2.07%)
Roll	1.3704° (6.85%)	1.3223° (2.47%)
Pitch	1.3650° (3.41%)	0.2289° (2.70%)
Yaw	0.3387° (3.39%)	0.1941° (4.84%)

motion model, specifically emulating the USS Navy ship Oliver Hazard Perry Class Frigate at sea state 6 with a 60-degree wave direction[43]. Given that the simulation model is smaller than the actual ship, adjustments in system sizes and motion amplitudes are necessary[44]. The position amplitudes (surge, sway, and heave) were scaled down with a 1/100 ratio, while maintaining the same magnitude of rotation. Figure 4.4b showcases the tracking error in the ship model simulation, displaying a position error ranging from -4.5×10^{-3} to 5.9×10^{-3} meters and a rotation error spanning from -2.91 to 3.14 degrees.

The root mean square errors (RMSE) and the percentage of normalized RMSE(NRMSE) for both motions are presented in Table 4.2. The difference between the maximum and minimum values of the desired trajectory served as a scaler to normalize RMSE. The results demonstrate that the CLF-QP accurately tracks the desired motions, and the platform state remains stable after the motion. At time $t = 5$, a drone landed on the moving platform, adding 1kg of additional mass to the platform. The RMSE of the tracking error without the drone landing was equivalent to Table 4.2, indicating that the CLF-QP controller maintained its performance after the drone landing on the platform.

Cable Tensions

Figure 4.5 illustrates the changes in cable tension as the moving platform undergoes motion. In both simulation scenarios, cable tensions were optimized without encountering optimization violations. Tensions remained within the predefined boundaries established before

initiating the simulation. During the periodic motion, lower cable tensions were consistently maintained within the range of $2 \sim 22.98$ N, while upper cable tensions fluctuated within $3.45 \sim 27.13$ N. Between the 2 and 4 seconds, two of the lower cables reached the minimum tension limit. However, the controller successfully identified tension solutions that stayed within the specified limits, ensuring that tensions for all cables remained within the desired boundaries. In the ship model motion, lower cable tensions varied within $4.96 \sim 20.21$ N, and upper cable tensions changed within $14.48 \sim 20.13$ N. Notably, cable tensions did not approach the tension limits and were consistently maintained within the desired tension boundaries throughout this scenario.

CLF and Stability

As outlined in 2, if a positive function $V(x)$ satisfies the inequality $\dot{V}(x) < 0$, the system is stable at the equilibrium with the feedback control input. As depicted in Figure 4.6, the CLF $V(x)$ remained positive while the platform was in motion and eventually converged to zero in both cases. The time derivative $\dot{V}(x)$, defined by (2.4), was consistently negative throughout the entire process. This indicates that the system is stable with an optimal input $u = [\vec{\tau}]$ calculated using (2.7).

4.2 Hardware Test

4.2.1 Hardware Design

The hardware setup consists of a moving platform connected by eight cables with the same dimensions as the simulation model. These cables are manipulated by eight DC motors to control their length, and tension sensors are connected to each cable to monitor their tension. In the following section, we will explore the mechanical design, electronic components, and the data flow within the cable-driven boat motion simulator.

Table 4.3. CDPR-based Boat Motion Simulator Hardware

Hardware	Supplier	Model
Outer Structure	Mcmaster	1-inch 8020 frame
DC motor	Pololu	Gearmotor 24V (Ratio 150:1)
Motor Driver	Pololu	DRV-8874
Tension Sensor	CALT	DYMH-103 Load cell force sensor
Wireless Module	HiLetgo	NRF24L01
Encoder Reader	Robogaia	8-channel encoder reader
Motor control MCU	PJRC	Teensy 4.1
Sensor MCU	Arduino	Mega, Uno
Power Supply	AlloverPower	3-24V DC Power Supply Kit
Moving Platform	Mcmaster	6061 Aluminum Plate (9057K377)
Cable	Mcmaster	High-Strength Thread (8800K43)

Mechanical Design

The platform was designed in a compact form, primarily emphasizing evaluating the capabilities of the CLF-QP controller and optimizing tension. The hardware components of the cable-driven boat motion simulator are itemized in the following Table 4.3. The outer framework of the simulator is constructed using 1-inch size 8020 frames, with each side of the structure measuring 0.45m in length. A 360-degree rotational cable pulley is positioned at each corner of the structure to ensure proper cable alignment and tension measurement. The platform is a 0.1m rectangular aluminum cuboid with a height of 0.025m. The tension sensor signal amplifier, an Arduino Mega, and a wireless module are mounted underneath the platform. Figure 4.7 displays the electronic components on the platform's underside. The platform has a total mass of 0.858kg.

Electronics and Data Processing

Figure 4.8 illustrates the electronic components and data processing of the simulator. The power supply unit provides a 24-voltage power to drive the DC motors. The Arduino Mega and the wireless module are independently powered by a 5-voltage power source to collect and transmit the tension sensor data. The Arduino Uno, equipped with an 8-channel encoder reader, is tasked with gathering data from the DC motor encoders. A wireless module connects Arduino

Mega to Uno for receiving tension sensor data. Arduino Uno, as a data hub collecting encoder and tension sensor data, is connected to the Teensy 4.1 MCU via the wireless module to transmit collected data. The Teensy 4.1 MCU calculates PWM signals to actuate motors leveraging PID controllers.

4.2.2 Tension Sensor

Sensor Calibration

The tension sensor used for the simulator is depicted in Figure 4.9a. These sensors operate on the principle that a load cell generates a voltage signal in response to an applied force. It is well-established that the output signal from the load cell exhibits a linear correlation with the magnitude of the applied load to the sensor[45][46]. Consequently, the Least Squares (LS) method is suitable for calibrating these force sensors[47]. The linear relationship between the applied weight, denoted as X, and the load cell force sensor reading, represented as Y, can be expressed as follows[48],

$$Y = aX + b, \tag{4.1}$$

where (a,b) are parameters. Also, the sum of the error square with n numbers of data can be defined as,

$$\varepsilon = \sum_{i=1}^{n} \big(Y_i - (aX_i + b)\big)^2. \tag{4.2}$$

To find parameters (a,b) to minimize (4.2), take the derivative of (4.2) about a and b,

$$\begin{aligned} \frac{\delta\varepsilon}{\delta a} &= 2a\sum_{i=1}^{n} X_i^2 + 2b\sum_{i=1}^{n} X_i - 2\sum_{i=1}^{n} X_i Y_i = 0, \\ \frac{\delta\varepsilon}{\delta b} &= 2a\sum_{i=1}^{n} X_i + 2nb - 2\sum_{i=1}^{n} Y_i = 0. \end{aligned} \tag{4.3}$$

Table 4.4. Tension Sensor Raw Data about Applied Calibration Weight

Sensor	0g	20g	50g	100g	200g	300g	500g	1000g
1	295597	292943	288598	281103	265507	250873	222293	149090
2	-12806	-14786	-17705	-20229	-30663	-37807	-56978	-96333
3	-16288	-17854	-20634	-25088	-33910	-42932	-58338	-100289
4	60755	57698	54046	47584	34290	21500	-5050	-131510
5	266313	262953	257772	251595	239787	224784	196951	127345
6	4960	3340	550	-4718	-11018	-20142	-35164	-75983
7	23130	20859	18479	11583	5284	-6556	-18530	-59979
8	197653	195719	193555	191132	184351	175049	151909	106794

The parameters are computed by solving (4.3) and written as,

$$a = \frac{n\sum X_i Y_i - \sum X_i \sum Y_i}{n\sum X_i^2 - \sum X_i \sum X_i}$$
$$b = \frac{\sum Y_i - a\sum X_i}{n}. \tag{4.4}$$

Calibration was conducted as shown in Figure 4.9b, utilizing a set of 11 calibration weights ranging from 0g to 1000g, and a part of the sensor's raw data is presented in Table 4.4. By leveraging this raw data from the sensors, the parameters denoted as 'a' and 'b' in (4.1) can be computed for each sensor using (4.2) through (4.4). The resultant parameters of the Least Square (LS) model for each tension sensor are displayed in Table 4.5. Figure 4.10 illustrates the LS model graph for the tension sensors, indicating a close-to-linear relationship between the raw data from the sensors and the corresponding calibration weight applied. The Root Mean Square Errors (RMSEs) between the LS model and the actual sensor raw data are summarized in Table 4.5.

Sensor Noise

Since each tension sensor has a different sensitivity, a sensor noise measurement was conducted to examine the sensor characteristics. The outcomes are depicted in Figure 4.11, and the corresponding standard deviation was computed. The sensor noise test involved collecting

Table 4.5. Tension Sensor Calibration Parameters and RMSE

Sensor	a	b	RMSE
1	-146.42	1443782	4.34g (0.04N)
2	-79.16	587084	10.74g (0.11N)
3	-85.69	476585	2.90g (0.03N)
4	-134.15	1495753	2.59g (0.03N)
5	-139.85	890682	2.09g (0.02N)
6	-81.53	220441	3.76g (0.04N)
7	-86.67	857388	7.94g (0.08N)
8	-79.05	787862	3.43g (0.03N)

10,000 samples for each sensor reading at a frequency of 10Hz without applying weights.

4.2.3 Control Algorithm

Figure 4.12a shows the control algorithm flow chart of the hardware model. The desired tension ($\vec{\tau}_m$) is calculated by the Control Lyapunov Function - Quadratic Program (CLF-QP) method leveraging CDPR dynamics depicted in chapter 2. Cable tension data ($\vec{\tau}$) measured by tension sensors is the input of the feedback system. The difference between the desired and the measured tension is used for the PID controller.

4.2.4 Results

We conducted tests in 2 different desired tension updates frequenciess, 1Hz and 10Hz. These frequency indicates the tension updates periods is 1 second, and 0.1 second. The test videos can be seen via the link:

1Hz Frequency Tension Test: https://youtu.be/Ys2Zgh4OTfY

10Hz Frequency Tension Test: https://youtu.be/ihPnC5amwp0

Tension tracking errors

The first data obtained from the tests comprised tension measurements recorded by tension sensors, as depicted in Figure 4.13. For the 1Hz test, the Root Mean Square Error

(RMSE) and Normalized RMSE (NRMSE) were 1.93N (12.40%) for the lower cable and 1.62N (15.44%) for the upper cables. Conversely, in the 10Hz test, the RMSE and NRMSE were 2.90N (28.09%) for the lower cable and 1.67N (23.60%) for the upper cables. Notably, the tension data update at 10Hz exhibited relatively larger errors compared to the 1Hz frequency update test. This discrepancy arises because the desired tension updates occur more rapidly at higher tension update frequencies than the actual motion actuation. Consequently, the desired tension is updated before the motors can drive the tension to the desired level and stabilize it, resulting in tracking delays and larger errors in tension tracking.

Platform State errors

To capture the platform state data, we employed a motion capture system. As illustrated in Figure 4.13, five motion capture markers were affixed to the moving platform, while three cameras were positioned to track their motion. These markers collectively formed a solid body, enabling us to derive the platform's position and rotation data from the camera-captured information. The motion capture video can be viewed via the following link: https://youtu.be/YF35A7JvgnY

Figure 4.15 presents the platform state errors. In the 1Hz tension update scenario, the RMSE (NRMSE) values were calculated as follows: Surge 0.48cm (16.17%), Sway 1.50cm (18.85%), Heave 2.09cm (17.70%), Roll 4.41°, Pitch 2.09° (17.23%), and Yaw 1.25° (17.76%). Conversely, in the 10Hz tension update case, the RMSE (NRMSE) values were as follows: Surge 0.67cm (28.45%), Sway 2.30cm (31.12%), Heave 3.50cm (40.08%), Roll 13.82° (37.94%), Pitch 3.07° (23.71%), and Yaw 1.60° (14.07%).

4.3 Analysis

The test results indicate that the desired tension computed utilizing CLF-QP methods with CDPR dynamics drives the platform toward the desired state, which is particularly evident in the low-frequency test. During slower motor operation, the PID controller is afforded sufficient time to track and stabilize the desired tension, resulting in relatively minor tension tracking errors

compared to high-frequency tests. Consequently, platform state tracking is notably accurate, especially with minimal errors observed in roll control. However, faster motor operation makes larger errors apparent compared to the low-frequency tension update test. The RMSE data for the higher frequency test underscores increased tension tracking errors. In scenarios of high frequency, tension updating outpaces motor actuation, leading to the next desired tension being updated before motors can reach and stabilize at the current desired tension. Consequently, delays in tension tracking ensue, resulting in substantial errors in the platform state.

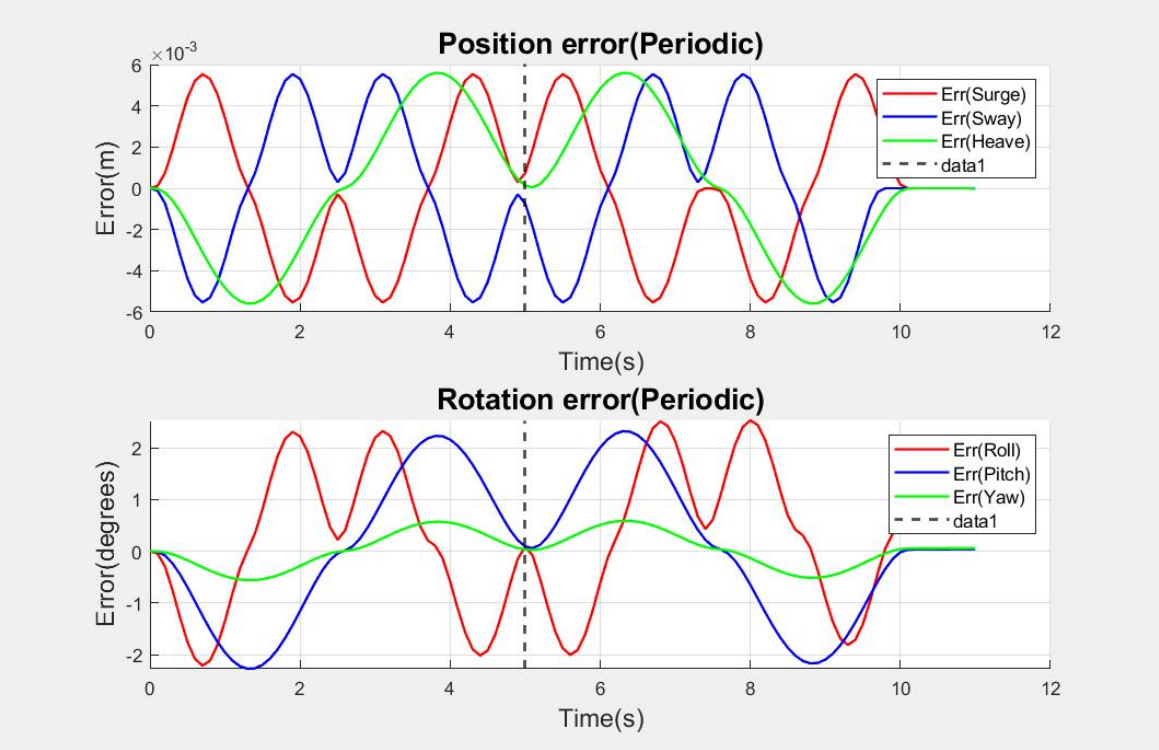
These tests underscore the need for hardware upgrades. The current model, featuring very stiff cables, is sensitive to even minor length changes, necessitating a more rigid structure and cable connections to handle high-tension situations effectively. Moreover, scaling up the hardware model offers benefits in error mitigation, as the system becomes less sensitive to cable length changes. Furthermore, upgrades are required for the motors, motor drivers, and power source. The existing motors demonstrate unreliable response to PWM signals, leading to unexpected motor behaviors, particularly under high tension. Therefore, replacing high-torque DC motors with a stable power source and robust motor drivers is essential to enhance system performance and reliability.

In the control domain, the PID controller exclusively relies on tension data, while encoder values tend to have less error compared to tension data. By utilizing encoder data, we can measure cable changes and determine the current cable length. Utilizing the known stiffness of the cables, cable tension can be computed using encoder data. Integrating both types of tension—directly measured by tension sensors and calculated from encoder data—holds promise for improving tension measurement accuracy. This integration can be facilitated through the implementation of complementary filters or Kalman filters. Studying this algorithm alongside hardware upgrades is imperative to construct a larger simulator.

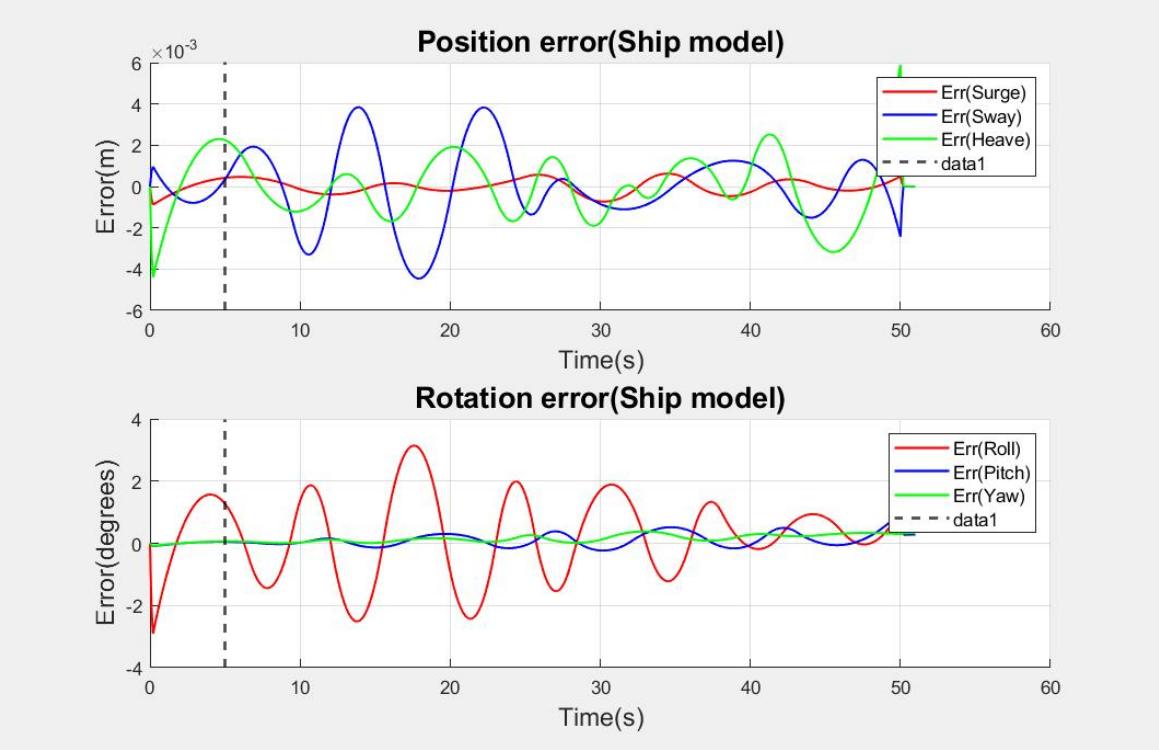
4.4 Acknowledgements

Support for this work was provided by Republic of Korea(ROK) Navy Sponsored Education Program.

Chapter 4, in part, is currently being prepared for submission for publication of the material, J. Jang, T. Bewley. The dissertation author was the primary investigator and author of this paper.

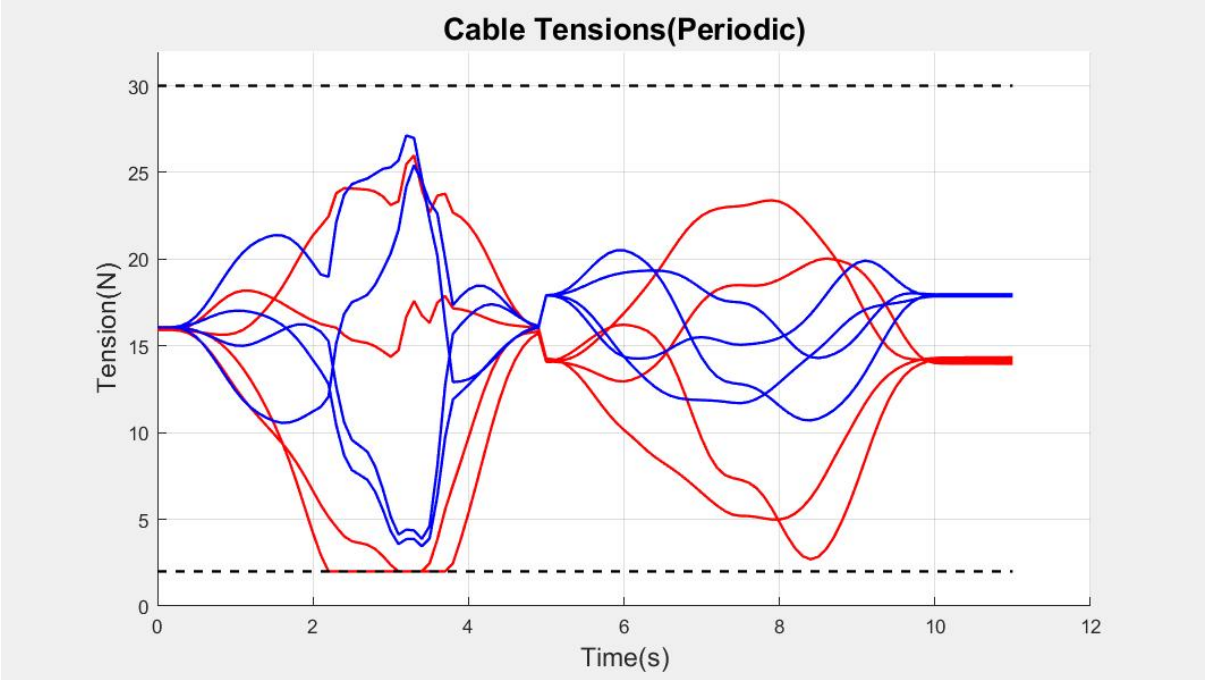


(a) Periodic motion

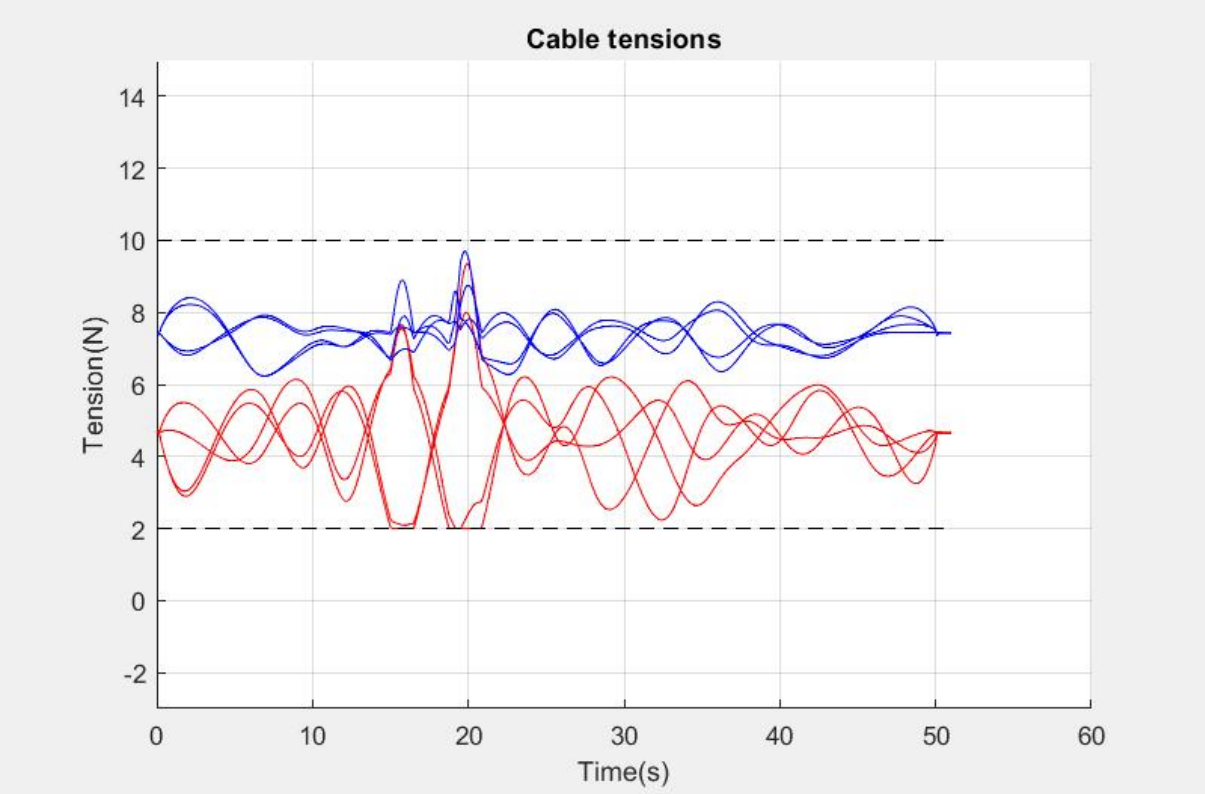


(b) Ship model motion

Figure 4.4. Motion tracking errors. The black dashed vertical lines are the time $t = 5$ when a drone lands on a platform. Controller constants were set as $p = 0.01$, and $\lambda = 1$.

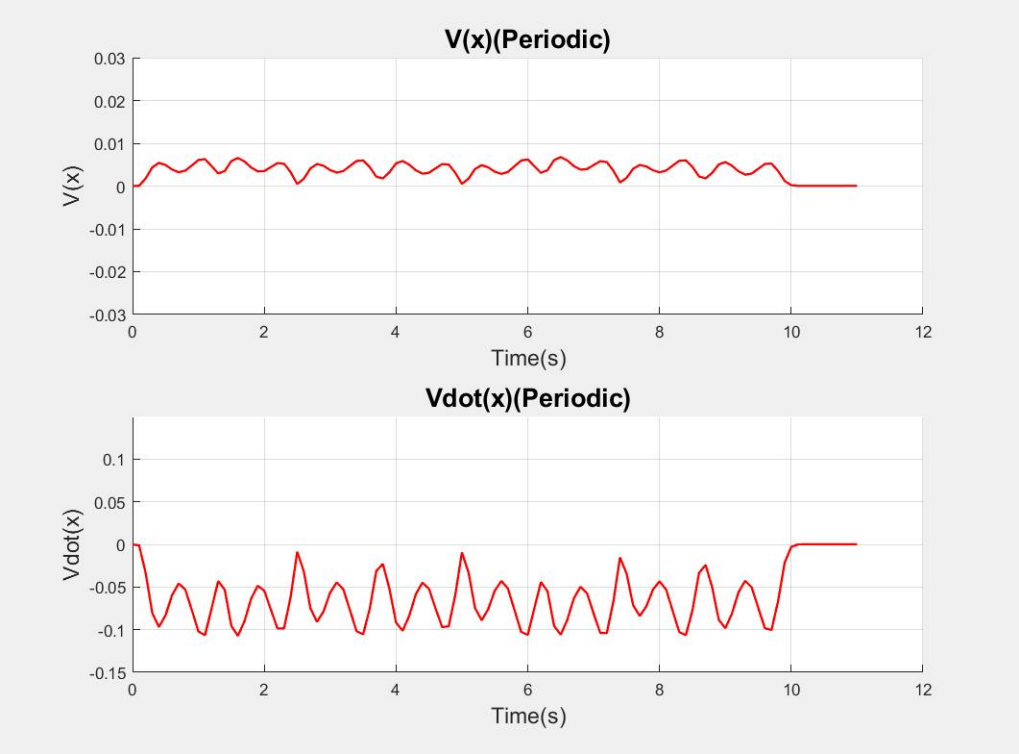


(a) Periodic motion

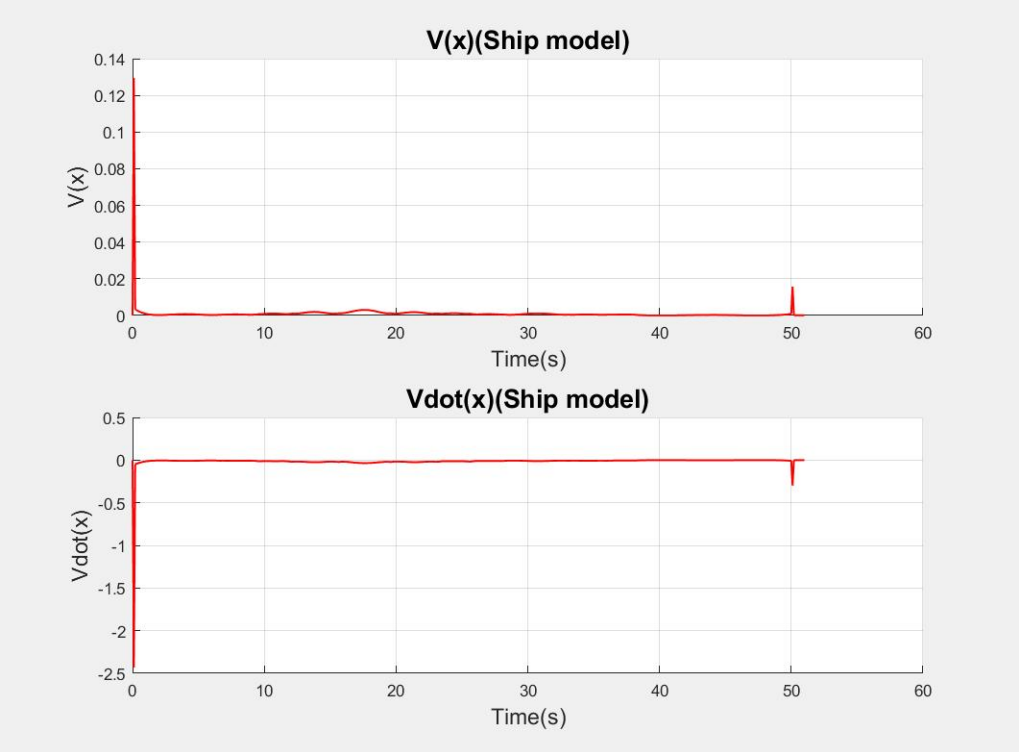


(b) Ship model motion

Figure 4.5. Cable tensions. The red lines are lower cables, and the blue lines are upper cables. Black dashed lines are the minimum and the maximum tension limits.



(a) Periodic motion: $V(x)$ is within the boundary $0 \sim 0.0063$, and $\dot{V}(x)$ is $-0.1067 \sim 0$.



(b) Ship model motion: $V(x)$ is within the boundary $0 \sim 0.1295$, and $\dot{V}(x)$ is $-2.4343 \sim 0$.

Figure 4.6. Control Lyapunov Function $V(x)$, and $\dot{V}(x)$

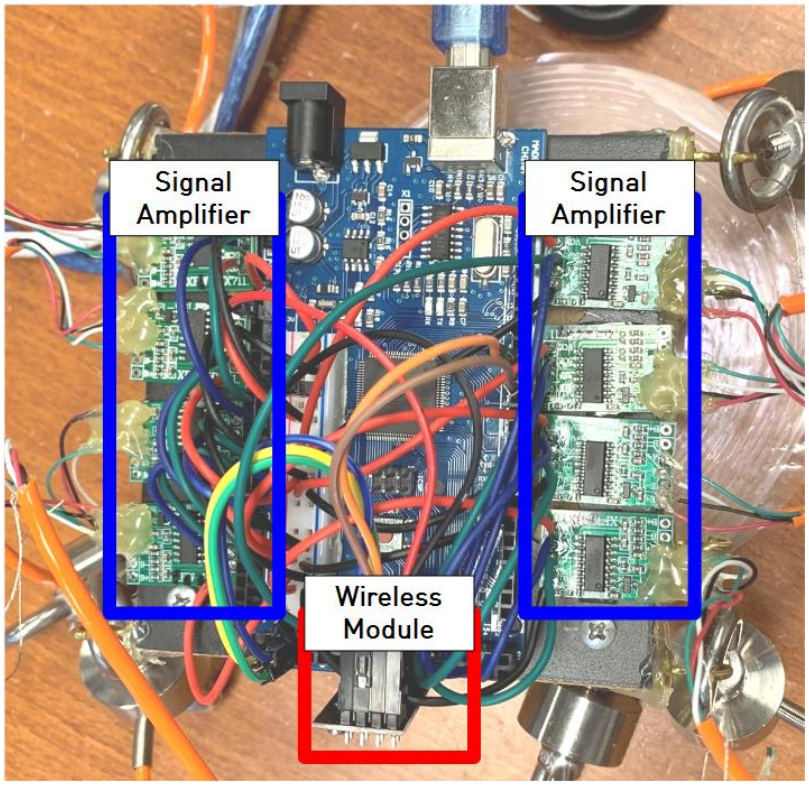


Figure 4.7. The electronics attached at the bottom of a moving platform. Signal Amplifiers(HX711) amplify tension sensor signals. The data is sent to Arduino Uno through a wireless module(NRF24L01).

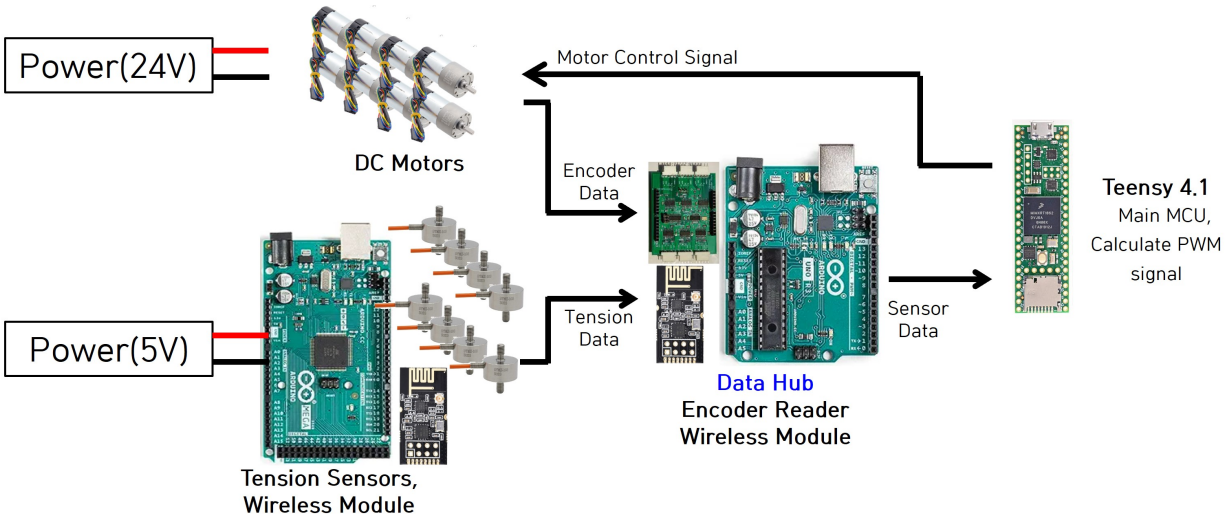
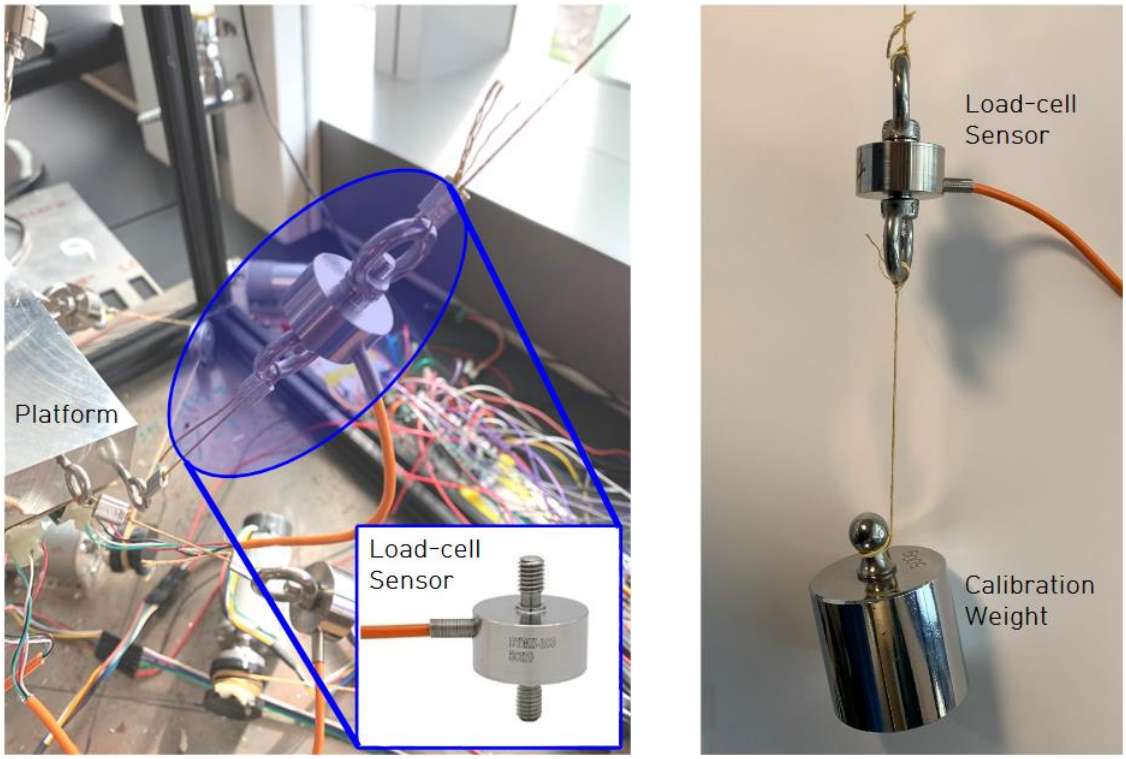
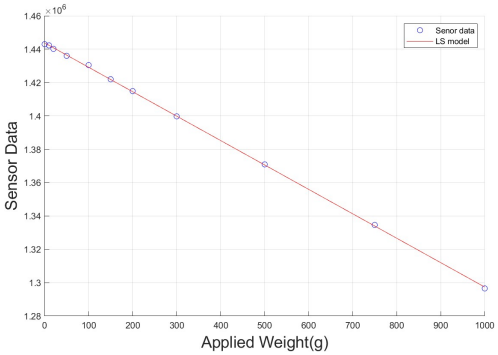


Figure 4.8. Electornics and Data Flow of the Cable-driven Robot Based Boat Motion Simulator

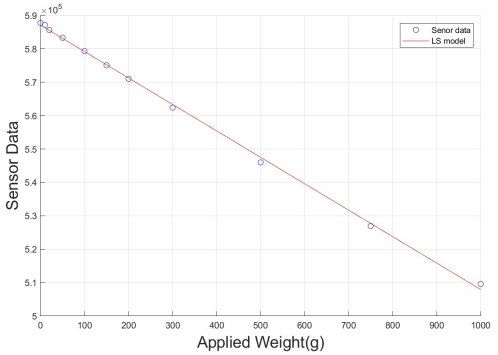


(a) Load Cell Tension Sensor attached to a platform **(b)** Tension Sensor Calibration

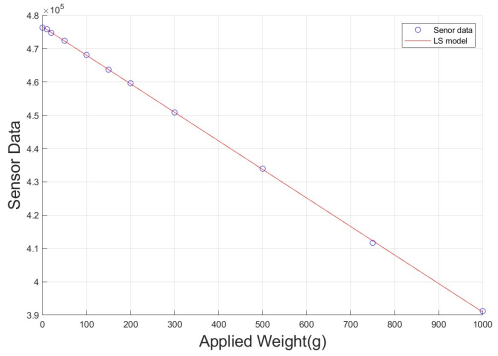
Figure 4.9. Load Cell Tension Sensor and its Calibration



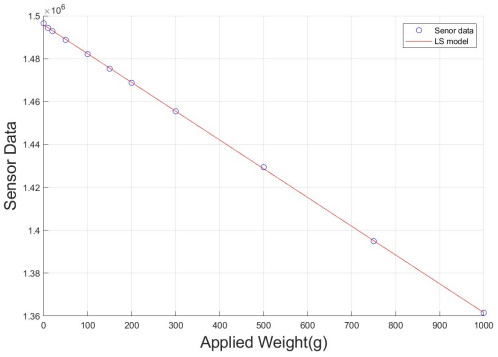
(a) Tension Sensor #1



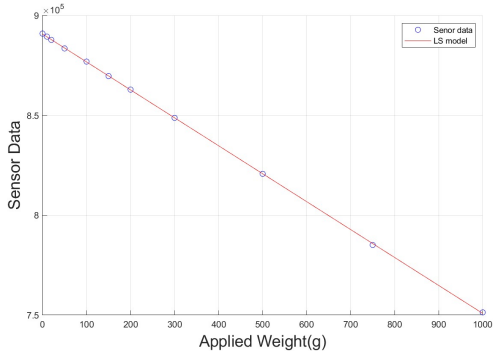
(b) Tension Sensor #2



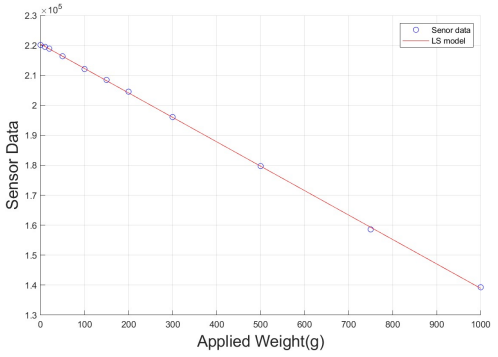
(c) Tension Sensor #3



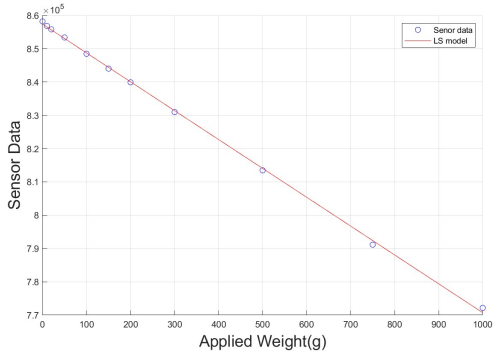
(d) Tension Sensor #4



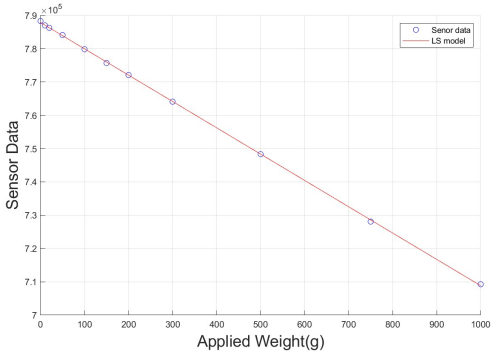
(e) Tension Sensor #5



(f) Tension Sensor #6

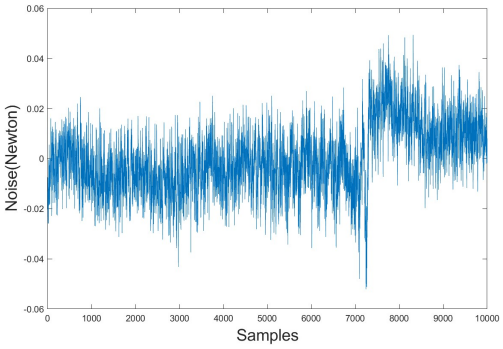


(g) Tension Sensor #7

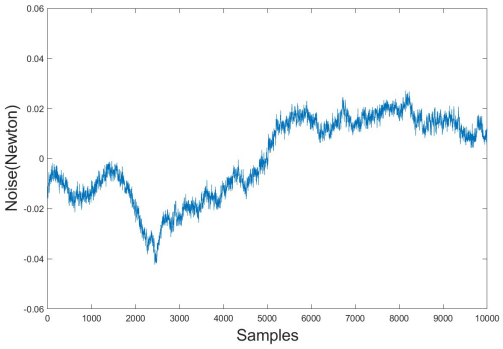


(h) Tension Sensor #8

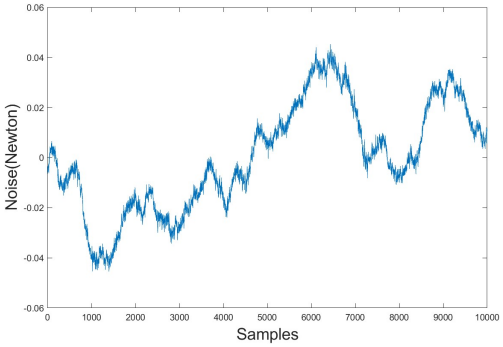
Figure 4.10. Tension Sensor Calibration Plot (Least Square Method).



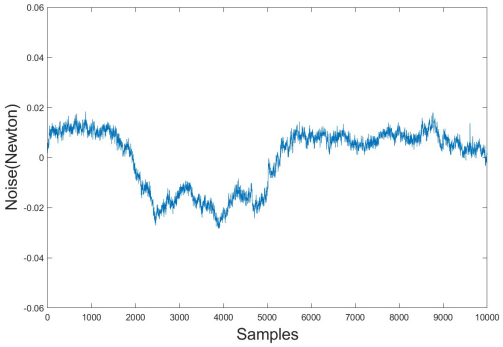
(a) Tension Sensor #1 ($\sigma_1 = 0.0132$)



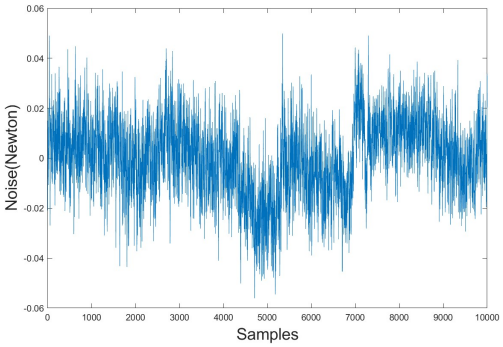
(b) Tension Sensor #2 ($\sigma_2 = 0.0163$)



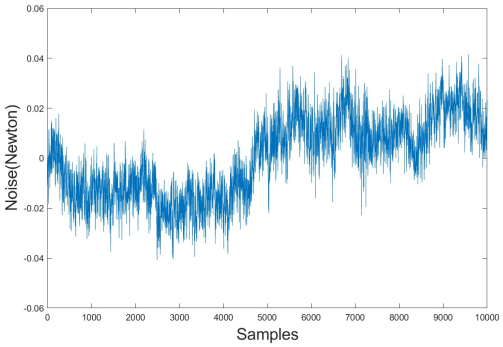
(c) Tension Sensor #3 ($\sigma_3 = 0.0214$)



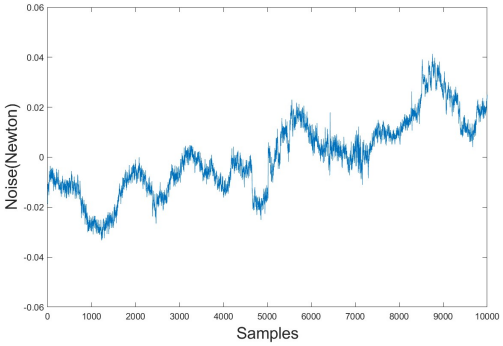
(d) Tension Sensor #4 ($\sigma_4 = 0.0119$)



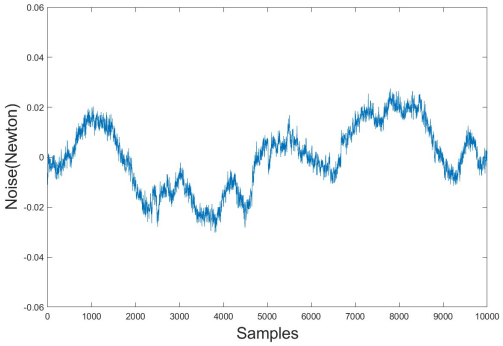
(e) Tension Sensor #5 ($\sigma_5 = 0.0149$)



(f) Tension Sensor #6 ($\sigma_6 = 0.0158$)

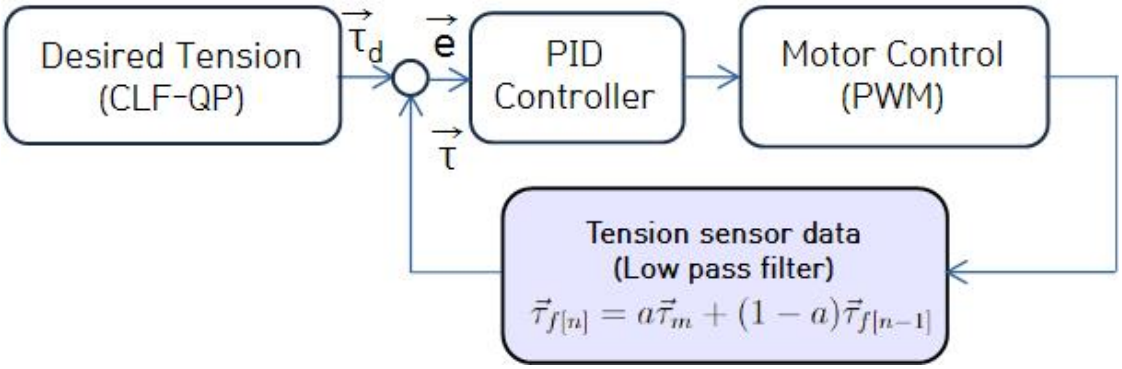


(g) Tension Sensor #7 ($\sigma_7 = 0.0150$)

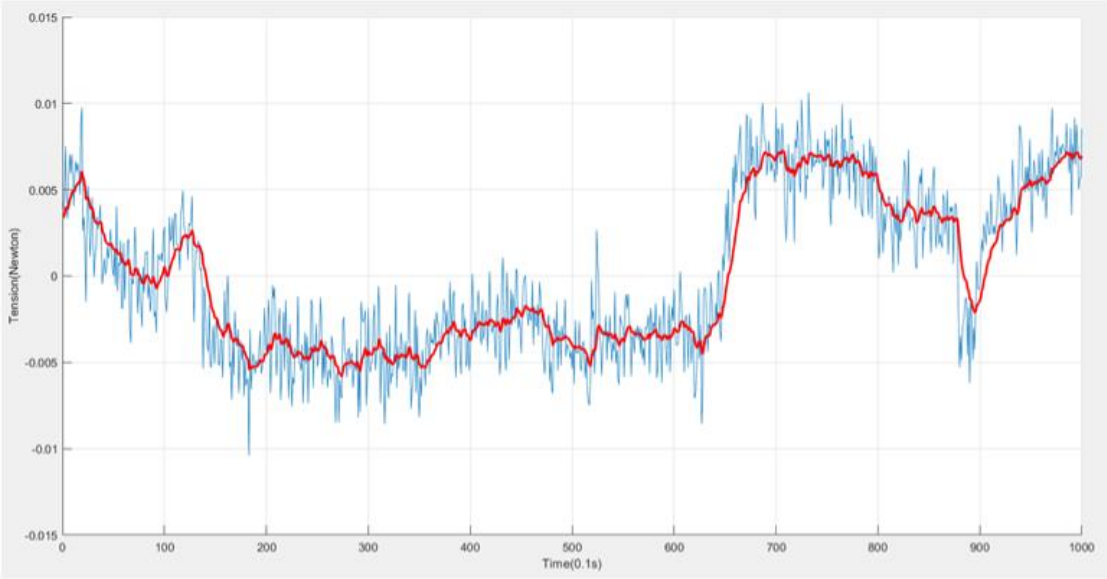


(h) Tension Sensor #8 ($\sigma_8 = 0.0130$)

Figure 4.11. Tension Sensor Noise Plot and Standard deviation σ (10000 data samples).

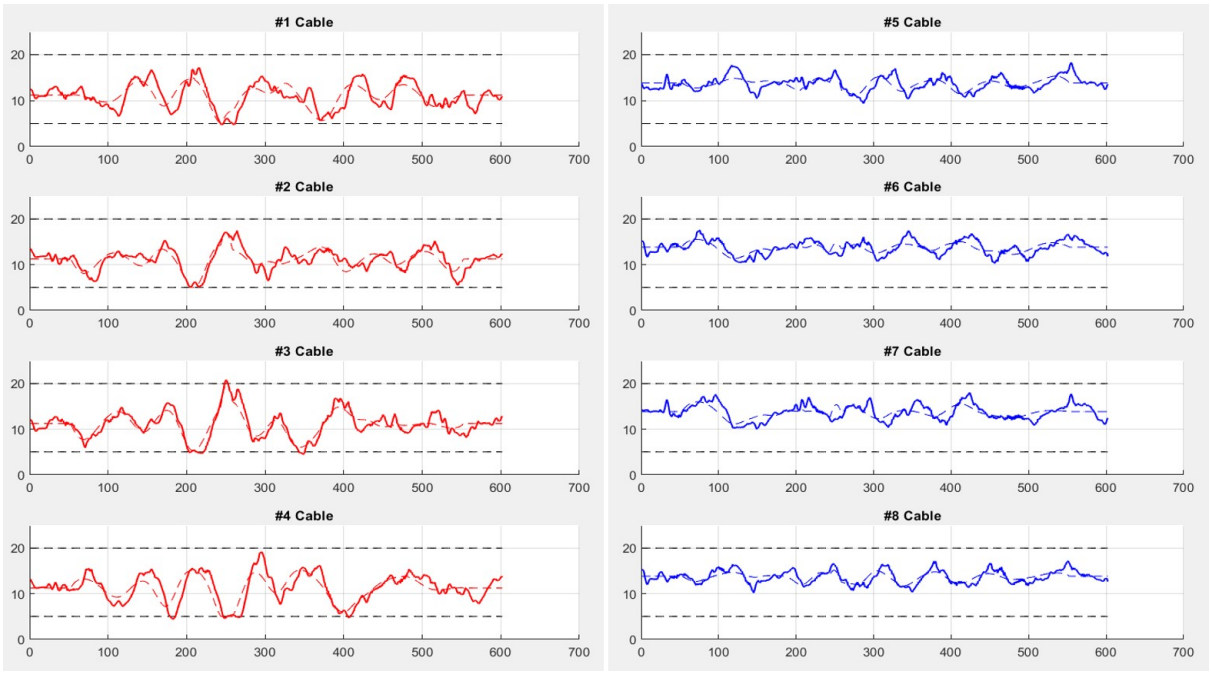


(a) Control Flow Chart of Cable-driven Boat Motion Simulator

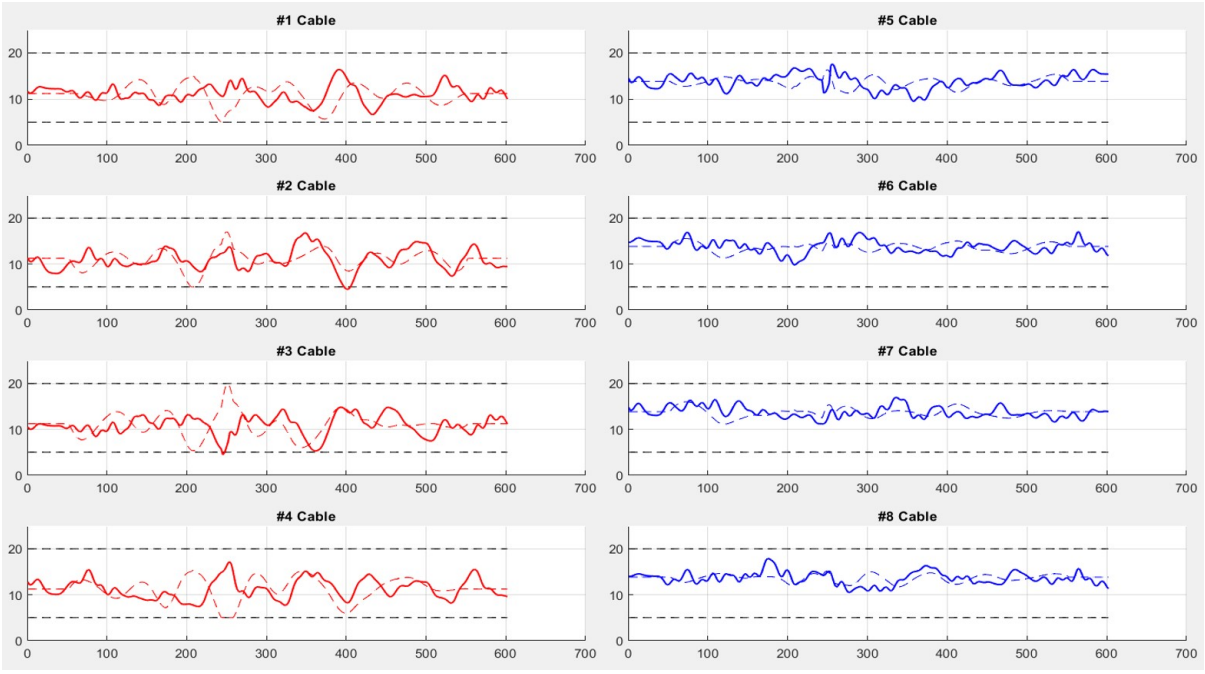


(b) Low pass Filter on #1 Tension Sensor Data (Blue: Raw data, Red: Filtered data)

Figure 4.12. Control Algorithm of Cable-driven Boat Motion Simulator and Applying the Low pass Filter on Tension Data.



(a) 1Hz Desired Tension Update



(b) 10Hz Desired Tension Update

Figure 4.13. Measured and Desired Tension Data (Red: Lower cables, Blue: Upper cables).

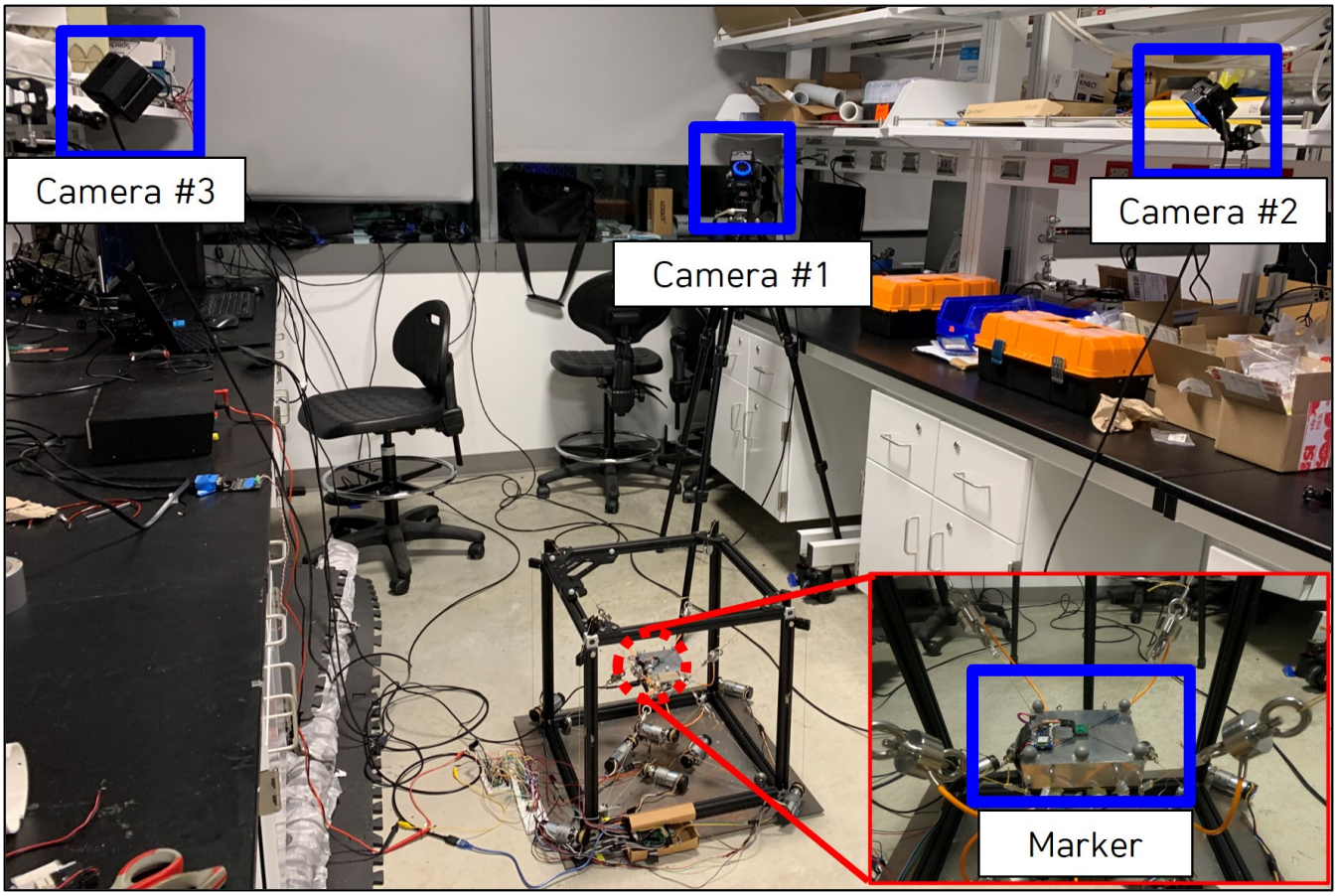
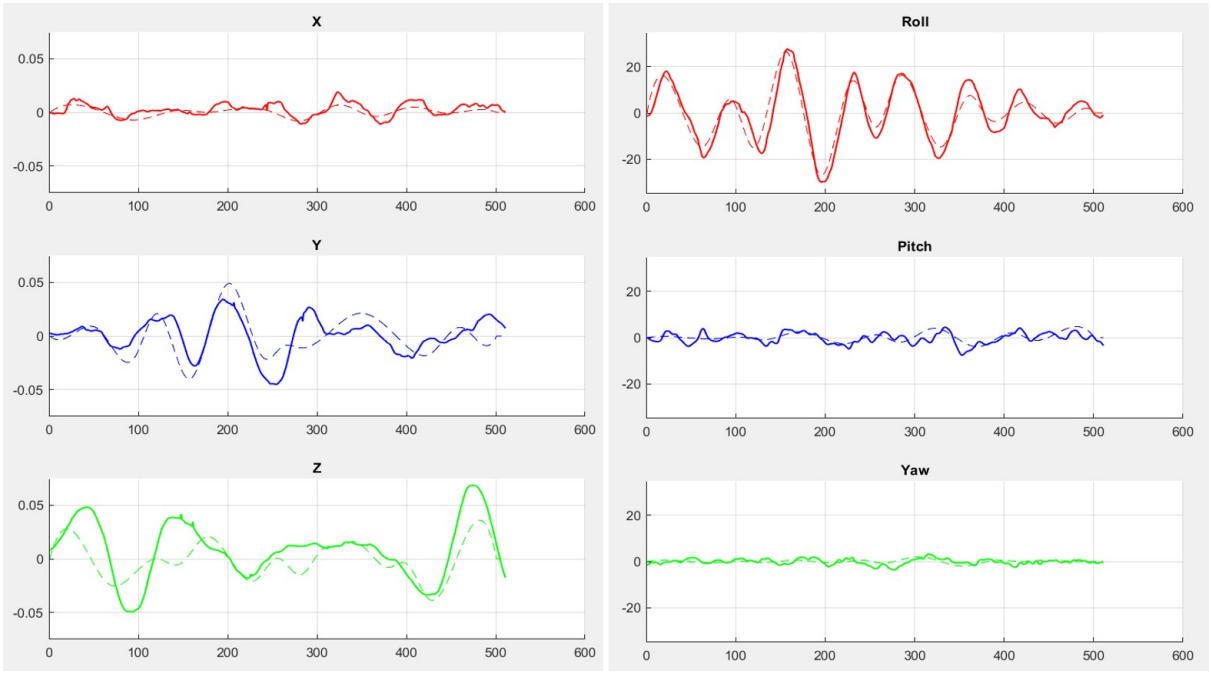
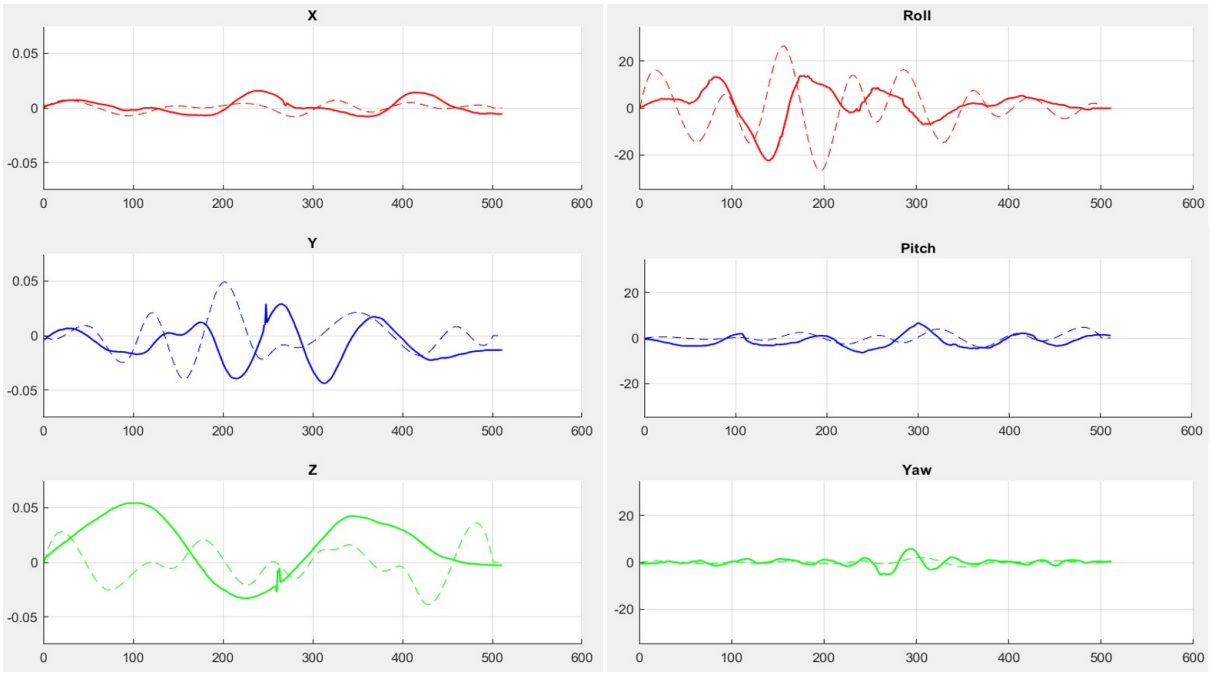


Figure 4.14. Motion Capture System Setup. 5 Markers were attached on a moving platform. 3 Cameras capture markers to calculate the position and rotation of a platform.



(a) 1Hz Desired Tension Update



(b) 10Hz Desired Tension Update

Figure 4.15. Measured(Dashed) and Desired(Solid) Platform State (Left: Surge(Top), Sway(Middle), Heave(Bottom), and Right: Roll(Top), Pitch(Middle), Yaw(Bottom).

Chapter 5

Camera Image Based Rotation Estimation for Quadrotor Landing

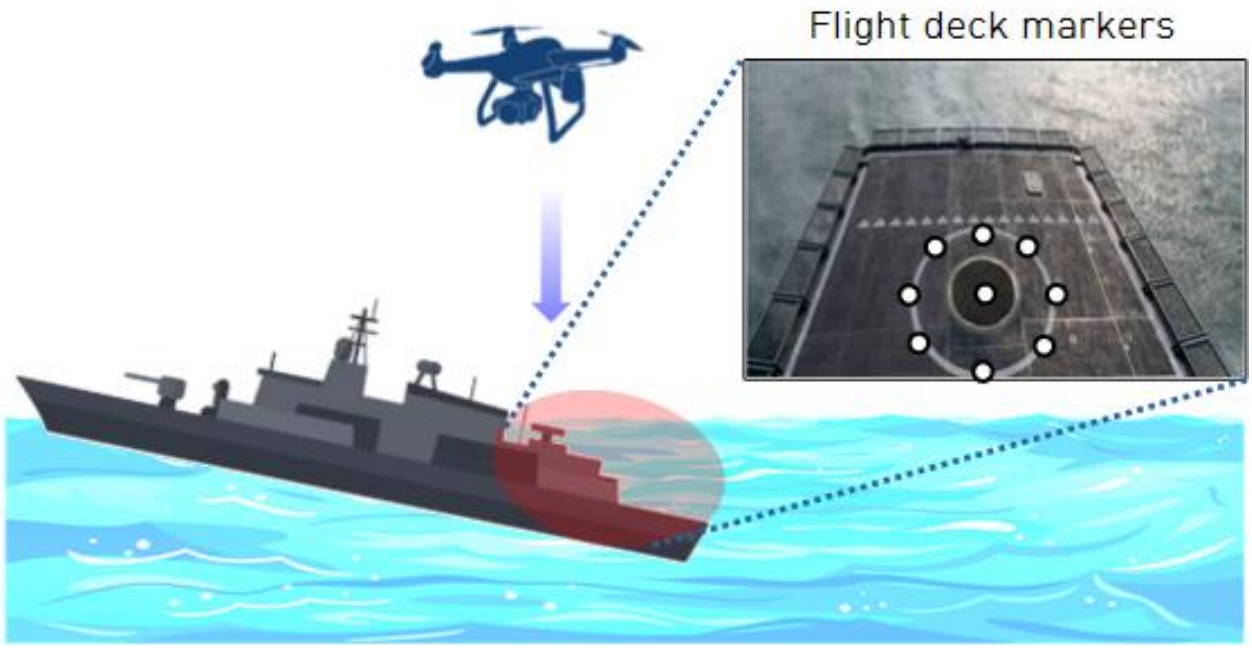


Figure 5.1. Conceptual image of a quadrotor landing on the navy ship after its mission. The circular pattern markers on the flight deck appear as an ellipse as the ship rotates

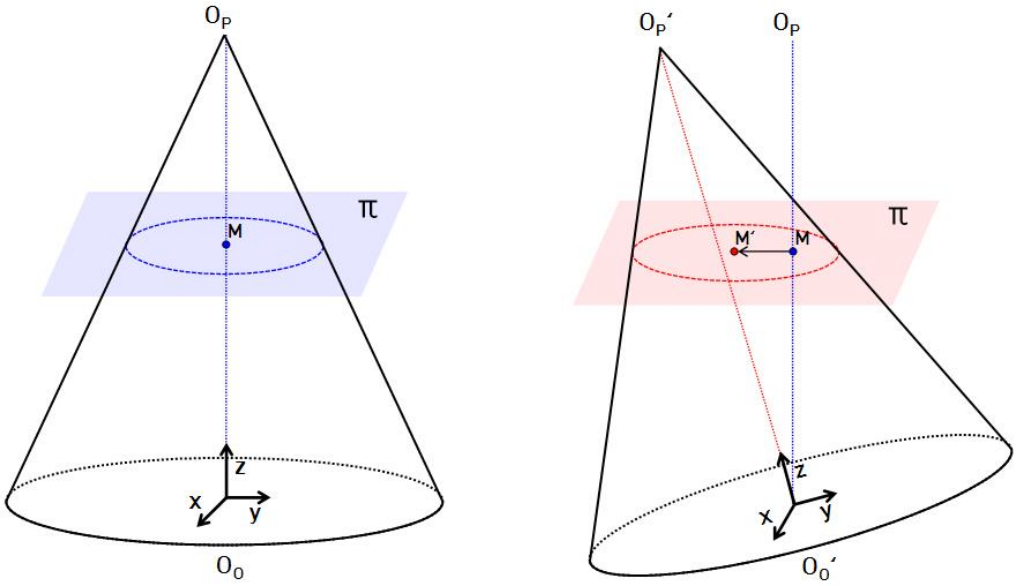
Estimating the 3D rotation of a moving platform using a 2D camera image is a valuable technique for safely recovering aerial vehicles onto the platform. Quadrotors and other Vertical Take Off and Landing Unmanned Aerial Vehicles (VTOL UAVs) face constraints on operation time due to their battery capacity. Consequently, these assets need to land on moving platforms

upon completing missions. There is a study focused on extending a quadrotor's operation time by utilizing a tethering power cable [51]. However, quadrotors inevitably need to return to the base platform for maintenance or to prepare for further operations. Platforms located on the sea, such as navy ships, experience 6-DOF motion during operations due to winds and waves. Inspired by the procedure of a navy helicopter landing on a navy ship's flight deck [52], we propose rotation estimation during the landing process of UAVs. In a related video, a navy helicopter pilot attempts to land on a navy ship in rough sea conditions. The pilot waits for the platform motion to stabilize sufficiently before safely landing on the flight deck.

Research studies on vision-guided autonomous UAV landing and the autonomous landing of aerial vehicles on ground vehicles primarily address precise landings on stationary or translationally moving targets [53][54]. However, these studies typically do not account for platform rotations. While there are studies discussing landing on a ship, they often focus on the estimation algorithm for the ship's position and do not take into consideration the ship's rotational motion [55]. Given that the ship's roll can reach approximately $\pm 30^{\circ}$ at sea state 6, there is a risk of causing damage to UAVs [56]. Consequently, a UAV may need to remain on standby at a safe altitude until the rotational motion is within a safe range for landing.

ArUco markers have been widely employed for estimating platform pose and facilitating the autonomous landing of UAVs [57][58][59]. While these markers aid in estimating rotations, deploying ArUco markers on the navy ship flight deck presents challenges. ArUco markers are conspicuous, making it easy for enemy reconnaissance aircraft to spot the navy ship. Additionally, the algorithm distinguishing ArUco markers may be rendered ineffective if the markers are damaged in combat situations. Therefore, we opted to explore the use of simple circle markers arranged in a circular pattern on the flight deck for rotation estimation.

We assume that a quadrotor is equipped with a camera capable of capturing 2D images and identifying markers on the flight deck, as depicted in Figure 5.1. This camera can recognize circular images and determine the pixel positions of the circles [60]. Initially, the centers of the camera image sensor and the circular pattern are aligned along the circular corn's revolution axis



(a) Zero rotation **(b)** 20° Roll

Figure 5.2. The conic section camera images as an object rotates. O_O is the circular shape object center, π is regarded as the camera image sensor, M is the center of the image on π, and O_P is the projection center.

[61]. As the object undergoes rotation, the contour of the image transforms into an ellipse, with a larger eccentricity. The ellipse equation can be established based on marker positions, and by analyzing the geometry of the ellipse, rotation estimation can be achieved [62]. A simulation model of the platform replicating ship motion and circular pattern markers was developed to simulate conic section camera images. As the platform rotated, circular pattern markers formed an ellipse, and the rotation estimation algorithm evaluated the rotation angles of the deck. Additionally, a hardware test was conducted to verify the performance of the method.

5.1 Preliminary

5.1.1 Conic Section Camera Image

Camera images reflect three-dimensional (3D) objects on a two-dimensional plane. In two-dimensional (2D) camera images, the object appears smaller as it moves farther from the camera, eventually converging at a single point, resulting in distortion known as the foreshorten-

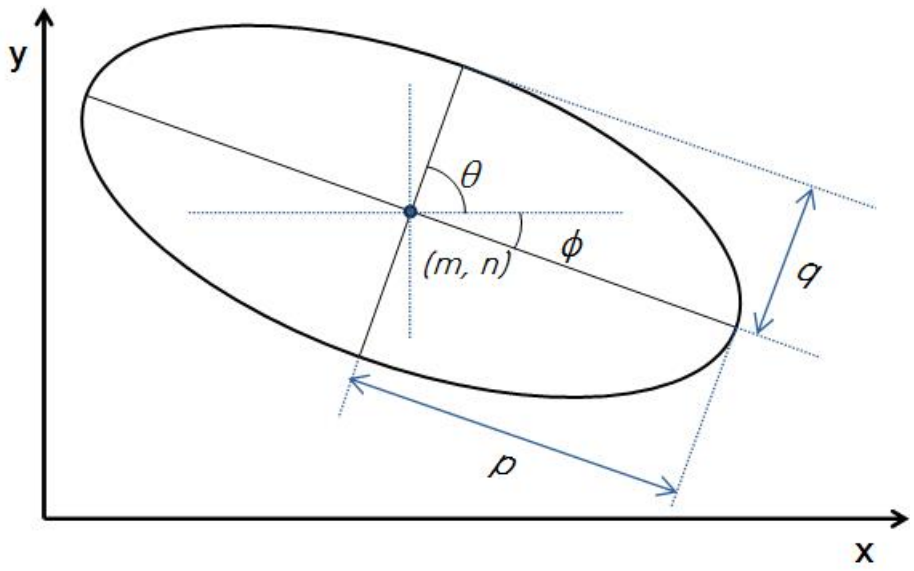


Figure 5.3. Ellipse with the center (m,n), semi-major axis p, and semi-minor axis q. θ, and ϕ are the tilt angles of an ellipse.

ing effect[63]. Figure 5.2 illustrates camera images on the image plane π based on the rotation of the object plane O_O. The camera image forms a conic section, resembling a cone formed by the revolution axis and a straight line connecting O_P and O_O. The shape of the projected images on π, represented by the blue and red dashed lines in Figure 5.2, transitions from a circle to an ellipse. Additionally, as the axis of the cone tilts, the center M of the circle shifts to the center M' of the ellipse[64][65]. Utilizing these properties, a camera model was developed and employed to simulate 2D camera images from 3D points. The camera model and conic section are defined by camera characteristics such as pixel resolution, focal angle, and focal length[66].

5.1.2 Ellipse Equation

Chapter 5.1.1 explains the changes in shape on the camera image plane as a circular object undergoes rotation. As the camera image transforms into an ellipse depending on the object's rotation, the estimation of rotation can be accomplished by analyzing the characteristics

of the ellipse. The equation of the ellipse, as illustrated in Figure 5.3, can be expressed as,

$$\frac{(x-m)^2}{p^2}+\frac{(y-n)^2}{q^2}=1, \tag{5.1}$$

where (m,n) represents the center of the ellipse position in the body frame, and (p,q) denotes the lengths of the semi-major and semi-minor axis, respectively. The ellipse equation can be reformulated in the polynomial form as follows[67],

$$ax^2+bxy+cy^2+dx+ey+f=0. \tag{5.2}$$

As (5.2) involves six unknown constants (a,b,c,d,e,f), a minimum of 6 positions (x,y) are required to determine the ellipse. By rearranging (5.2) and moving the term ax^2 to the right-hand side, the matrix form of the ellipse equation becomes,

$$\begin{bmatrix} x_1y_1 & y_1^2 & x_1 & y_1 & 1 \\ x_2y_2 & y_2^2 & x_2 & y_2 & 1 \\ x_3y_3 & y_3^2 & x_3 & y_3 & 1 \\ & & \vdots & & \\ x_iy_i & y_i^2 & x_i & y_i & 1 \end{bmatrix} \begin{bmatrix} b' \\ c' \\ d' \\ e' \\ f' \end{bmatrix} = \begin{bmatrix} -x_1^2 \\ -x_2^2 \\ -x_3^2 \\ \vdots \\ -x_i^2 \end{bmatrix}. \tag{5.3}$$

Here, b',c',d',e', and f' are derived as follows,

$$b'=\frac{b}{a}, \quad c'=\frac{c}{a}, \quad d'=\frac{d}{a}, \quad e'=\frac{e}{a}, \quad f'=\frac{f}{a},$$

where $a=1$, and (x_i,y_i) represents the position of i-th point on the ellipse, a minimum of five points are required to define the ellipse equation. Solving (5.3) enables the determination of five unknown constants. The resulting ellipse parameters (m,n,p,q) in (5.1) are expressed as

follows[68][69],

$$
\begin{aligned}
m &= \frac{cd - bf}{b^2 - ac}, \quad n = \frac{ae - bd}{b^2 - ac}, \\
p &= \sqrt{\frac{2(ae^2 + cd^2 + fb^2 - 2bde - acf)}{(b^2 - ac)(\sqrt{(a-c)^2 + 4b^2}) - (a+c)}}, \\
q &= \sqrt{\frac{2(ae^2 + cd^2 + fb^2 - 2bde - acf)}{(b^2 - ac)(-\sqrt{(a-c)^2 + 4b^2}) - (a+c)}}.
\end{aligned}
$$

The tilt angle of an ellipse, which indicates the angle between the minor axis and the horizontal axis of the image plane, can be formulated as follows[62][70],

$$
\theta = \begin{cases}
\frac{1}{2}\cot^{-1}(\frac{c-a}{2b}) + \frac{\pi}{2}, & \text{if } p < q \\
\frac{1}{2}\cot^{-1}(\frac{c-a}{2b}) - \pi, & \text{if } b < 0, \quad \frac{1}{2}\cot^{-1}(\frac{c-a}{2b}) > \frac{\pi}{2} \\
\frac{1}{2}\cot^{-1}(\frac{c-a}{2b}), & \text{otherwise}
\end{cases} \tag{5.4}
$$

and the range is $(-\pi/2 \leq \theta \leq \pi/2)$. Additionally, the angle(ϕ) between the major axis and the horizontal axis of the image can be expressed in terms of θ,

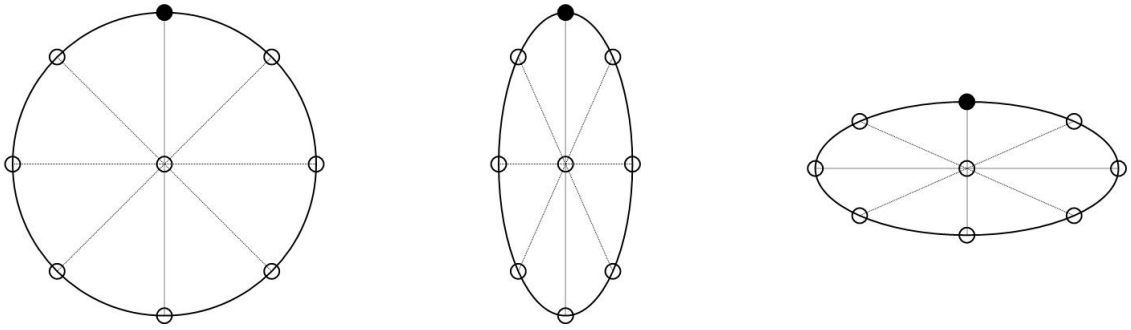
$$
\phi = \text{sign}(\theta)(\frac{\pi}{2} - \theta).
$$

5.1.3 Rotation Sequence

The 3D rotation of a solid body can be described by a set of three rotation angles (α, β, γ). Tait-Bryan and Euler rotation sequences are commonly used to represent the rotation of an object.

Sign Ambiguity

The estimation of rotation from a 2D image can lead to the rotation sign ambiguity problem. When the distance between the object and the camera is considerable, or the rotation is very small, the distortion in the camera image is negligible. Consequently, when a platform



(a) Zero rotation **(b)** Roll **(c)** Pitch

Figure 5.4. Top view of markers on a platform. Hollow circles represent markers, and the black circle denotes the bow marker. The markers form a circle with zero rotation but transform into an ellipse when the platform rotates. In cases of only roll or pitch, the direction of rotation becomes ambiguous.

rotates only in roll or pitch, the direction of rotation cannot be determined because the images appear equivalent, as illustrated in Figure 5.4.

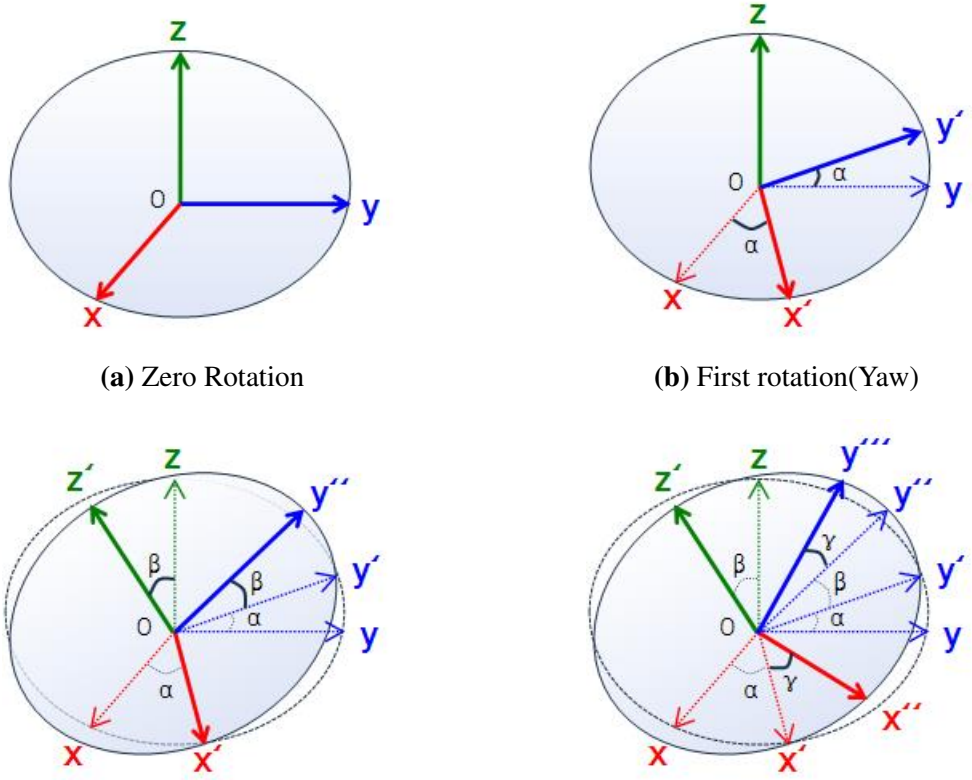
3-1-3 Euler Rotation Sequence

The 3-2-1 Tait-Bryan rotation sequence comprises yaw, pitch, and roll pitch angles. There are potentially two sign ambiguities, which involve roll and pitch. However, the 3-1-3 Euler rotation sequence consists of yaw, roll, and another yaw angle. Therefore, employing the 3-1-3 Euler angle rotation sequence is advantageous in reducing the possibility of sign ambiguity. As illustrated in Figure 5.5, the 3D rotation of an object can be described using the 3-1-3 Euler rotation sequence: Rotate the object by an angle α about the z-axis of the body frame, then by β about the x-axis, and finally by γ about the z-axis.

5.2 Rotation Estimation

5.2.1 Determine Camera Image Conic Section

Assume that there are nine markers on a ship flight deck. Eight markers are positioned in a circular pattern, and one marker is at the center of the circle. Figure 5.2 illustrates conic sections shown on the camera image plane as a ship rotates. As explained in 5.1.1, a rotated circle



(a) Zero Rotation

(b) First rotation(Yaw)

(c) Second Rotation(Roll)

(d) Third rotation(Yaw)

Figure 5.5. 3-1-3 Euler rotation sequence

image appears as an ellipse, and the ellipse's center is shifted. The pixel coordinates position (X_c, Y_c, Z_c) of each marker, representing its position relative to the camera coordinate system with the origin at the center, can be computed using the pinhole camera model equation[66].

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = K * T * \begin{bmatrix} P_x \\ P_y \\ P_z + h \end{bmatrix}, \tag{5.5}$$

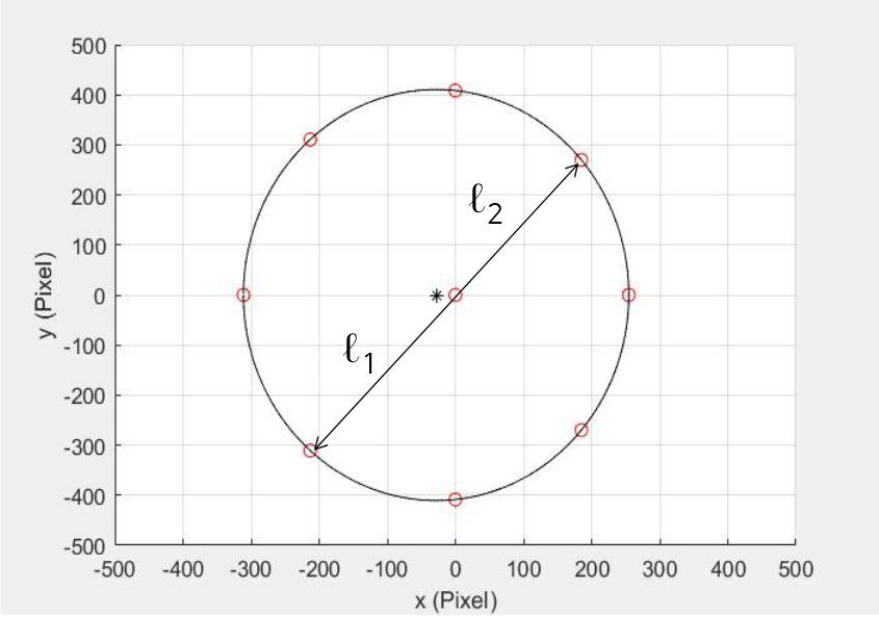


Figure 5.6. Simulated distorted camera image determined by 20◦ roll platform rotation. Red circles are circular pattern markers. The black star is the ellipse's center that is shifted by rotation.

where (P_x, P_y, P_z) is 3D marker position in platform coordinate and h is the altitude of quadrotor. Matrices K, T are defined as,

$$K = \begin{bmatrix} fx & 0 & cx \\ 0 & fy & cy \\ 0 & 0 & 1 \end{bmatrix}, \quad T = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix},$$

where fx and fy are focal lengths, cx and cy indicate the camera center coordinate. Matrix K is determined by camera characteristics, and matrix T represents the transformation from the object coordinate to the camera image coordinate. As the image coordinate system's origin is located at the top left corner, 2D-pixel marker positions (X_p, Y_p) can be written as,

$$X_p = \frac{X_c}{Z_c} - cx, \quad Y_p = cy - \frac{Y_c}{Z_c}. \tag{5.6}$$

Finally, by (5.3), the equation of the ellipse in polynomial form can be determined, given that the 2D-pixel marker positions are expressed by (5.6).

5.2.2 Obtain Corrected Image

As described in 5.1.1, camera images are distorted due to the foreshortening effect, and this effect can solve the sign ambiguity problem because the ellipse center is shifted. However, distorted camera images must be corrected since the rotation sequence illustrated in Figure 5.5 is based on geometrically undistorted images. The 2D i-th marker position vectors $\vec{l}_i = (X_{p_i}, Y_{p_i}) + \vec{l}_0$ with the center marker position $\vec{l}_0 = (X_{p_0}, Y_{p_0})$, can be defined as,

$$\vec{l}_i = d_i \vec{c}_i,$$

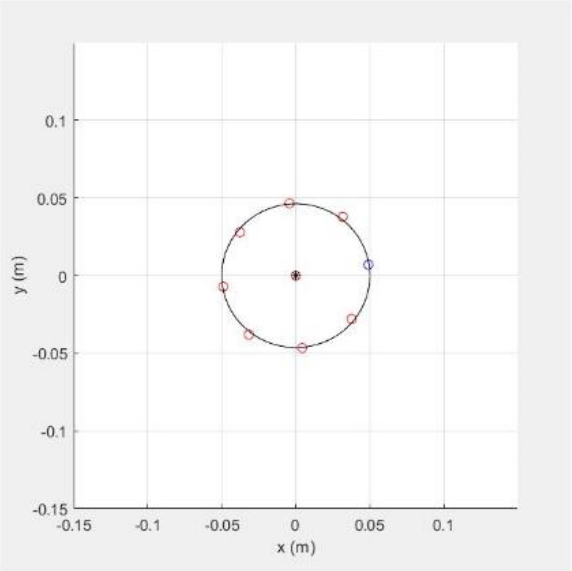
where $\vec{c}_i$ is the unit marker position vector, d_i is the distance between the center marker and the i-th marker. In the distorted image shown in Figure 5.6, $\vec{l}_1$ and $\vec{l}_2$ are marker position vectors in the opposite direction. Assuming that a quadrotor has an altimeter and maintains hovering at the center of the platform, the distorted camera image can be converted to the undistorted marker positions in the platform coordinate utilizing (5.5), (5.6). By plugging $\vec{l}_i$ into the pinhole camera model equation and rearranging equations, then the formulations are derived as below[62],

$$\begin{aligned} P_{x_i} &= \frac{-d_1 c_{x_0} - d_2 c_{y_0} - 2d_1 d_2 \cos(\Psi)}{fy(d_1 - d_2)} h \\ P_{y_i} &= \frac{d_1 c_{x_0} + d_2 c_{y_0} + 2d_1 d_2 \sin(\Psi)}{fx(d_1 - d_2)} h \\ P_{z_i} &= \frac{-d_1 - d_2}{d_1 - d_2} h, \end{aligned} \tag{5.7}$$

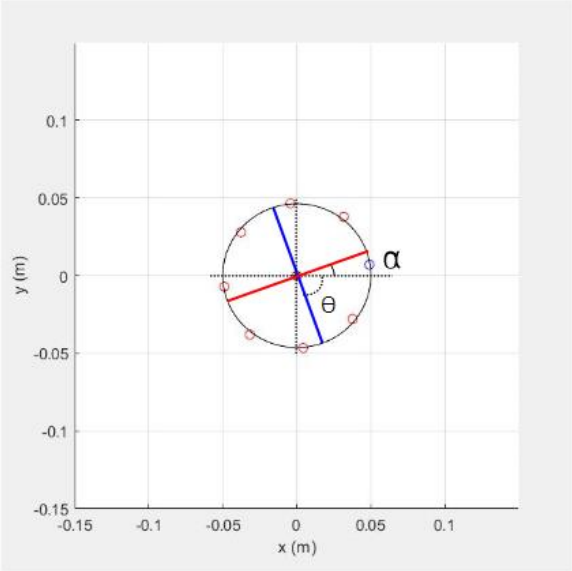
where (c_{x_0}, c_{y_0}) represents the unit center marker position vector, and Ψ is the angle of the marker position with respect to the center marker.

5.2.3 Rotation Estimation Equations

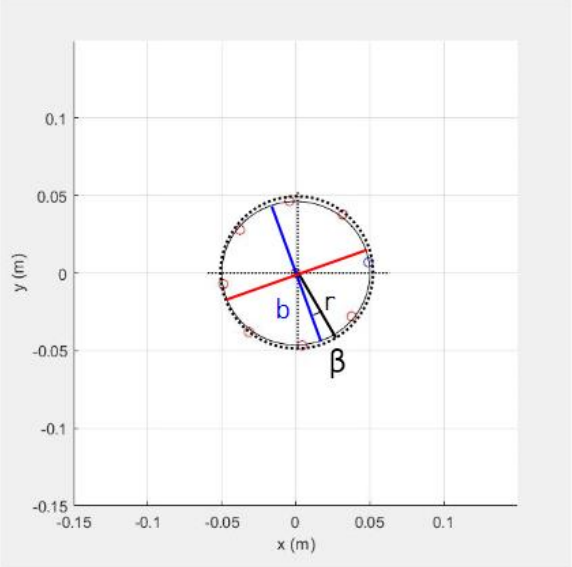
Figure 5.7 depicts the simulated undistorted images outlined in 5.2.2 and the procedure for determining the 3-1-3 Euler rotation sequence from the images. The detailed process is as



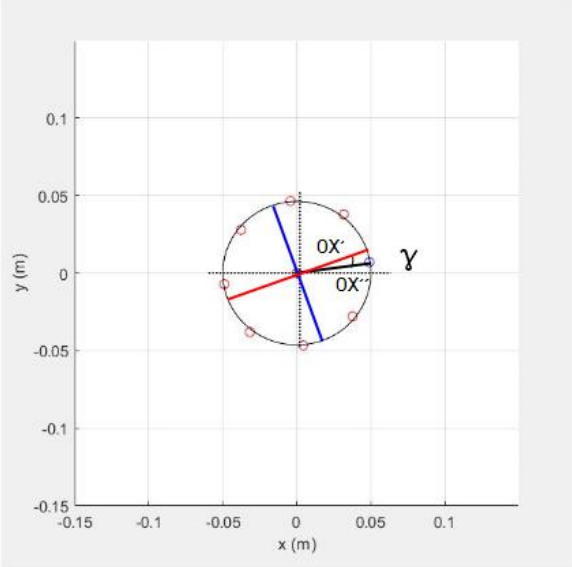
(a) Ellipse after rotation



(b) First rotation(Yaw)



(c) Second Rotation(Roll)



(d) Third rotation(Yaw)

Figure 5.7. 3-1-3 Euler rotation sequence angles from undistorted camera images for 20° roll, 10° pitch, and 5° yaw in the platform frame. The red line is the major axis, and the blue line is the minor axis.

follows[62]:

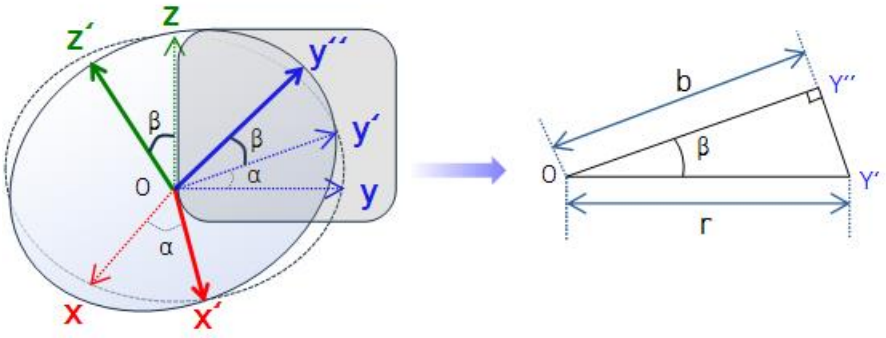


Figure 5.8. The side section view of the right triangle formed by the second rotation. *b* is the semi-minor axis length and *r* is the marker circle radius.

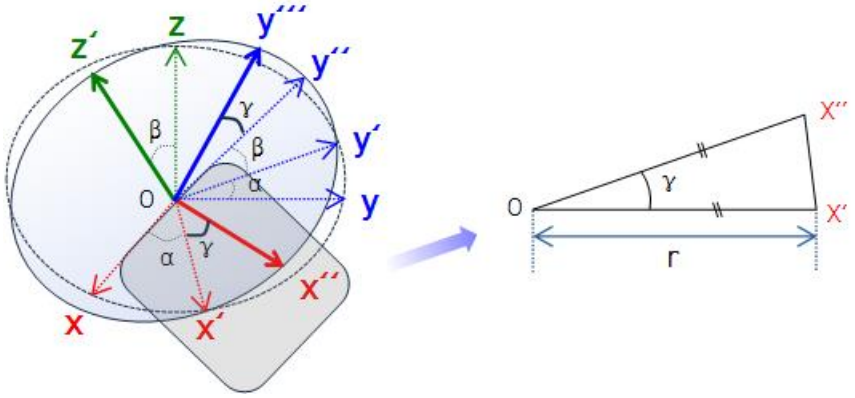


Figure 5.9. The angled side section view of the isosceles triangle formed by the third rotation.

The first rotation(Yaw)

The first rotation angle is the semi-major axis tilt angle for the horizontal axis. Figure 5.7b is graphically describing that α is the tilt angle of the ellipse and can be expressed as,

$$\alpha = \phi = \text{sign}(\theta)(\frac{\pi}{2} - |\theta|).$$

The second rotation(Roll)

As shown in Figure 5.7c and 5.8, β is the angle between the semi-minor axis and the radius vector of the marker circle. The side section of the semi-minor axis,and the radius vector forms a right triangle with rotation angle β. Since markers form a circle before the second

rotation, the longer side of the triangle is circle radius r. The sign of β is determined by the z-component of the semi-minor axis OY''. Finally, the equation for β can be formulated as,

$$\beta = \text{sign}(Y_z'')\cos^{-1}(\frac{b}{r}),$$

where b represents the length of the semi-minor axis.

The third rotation(Yaw)

The third rotation can be estimated by calculating the rotation angle between major axes OX' and the bow marker vector OX''. In Figure 5.7d, the length of the major axis is the same as circular pattern radius r, equivalent to the length of OX' and OX''. Figure 5.9 shows the triangle formed by the third rotation angle. Since the section is an isosceles triangle, γ can be calculated leveraging the law of cosines, and the sign of the angle is determined by the y-component of the bow light vector OX'', which can be determined by (5.7). Then, the formulation of γ becomes,

$$\gamma = \text{sign}(X_y'')(\cos^{-1}\big(\frac{2r^2 - |OX' - OX''|^2}{2r^2}\big).$$

Rotation conversion

Since the estimated rotation angles are in the 3-1-3 Euler rotation sequence, the rotations need to be converted to determine roll, pitch, and yaw angles. First, 3-1-3 Euler rotation angles (α, β, γ) calculated by the estimation process can be converted to quaternion rotations $q = [q_0, q_1, q_2, q_3]$ by,

$$\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} c_{\gamma/2}c_{\beta/2}c_{\alpha/2} - s_{\gamma/2}c_{\beta/2}s_{\alpha/2} \\ c_{\gamma/2}s_{\beta/2}c_{\alpha/2} + s_{\gamma/2}s_{\beta/2}s_{\alpha/2} \\ s_{\gamma/2}s_{\beta/2}c_{\alpha/2} - c_{\gamma/2}s_{\beta/2}s_{\alpha/2} \\ c_{\gamma/2}c_{\beta/2}s_{\alpha/2} + s_{\gamma/2}c_{\beta/2}c_{\alpha/2} \end{bmatrix},$$

with $c_{\alpha/2} = \cos(\alpha/2)$ and $s_{\alpha/2} = \sin(\alpha/2)$. Once the quaternion rotation is obtained, the roll, pitch, and yaw angles can be computed by the quaternion-Euler angle conversion formulation as follows[71],

$$\begin{aligned}
\text{Roll} &= \text{atan2}\big(2(q_0q_1 + q_2q_3), 1 - 2(q_1^2 + q_2^2)\big), \\
\text{Pitch} &= \text{asin}\big(2(q_1q_3 - q_0q_2)\big), \\
\text{Yaw} &= \text{atan2}\big(2(q_0q_3 + q_2q_1), 1 - 2(q_2^2 + q_3^2)\big).
\end{aligned}$$

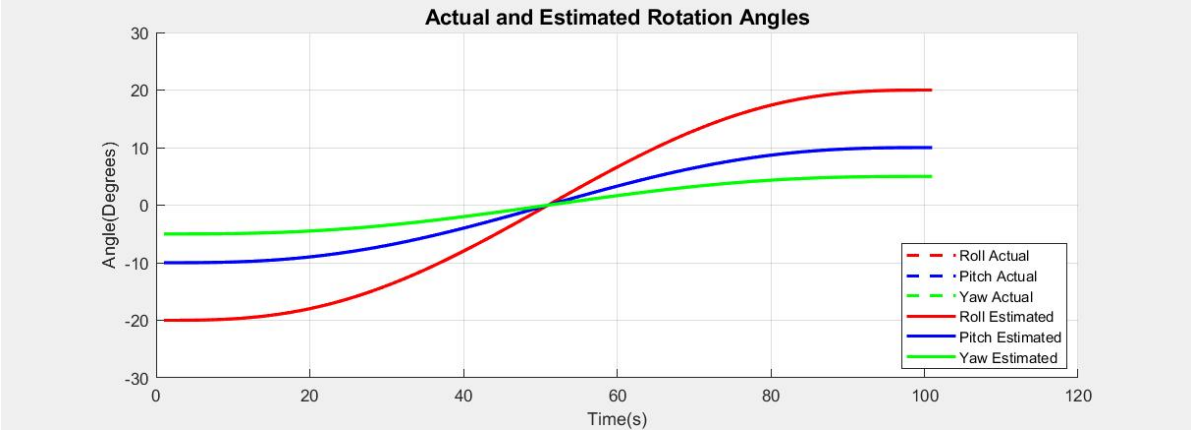
5.3 Experiments

To validate the theory, a simulation model was developed and tested. Additionally, a scaled-down hardware experiment was conducted to assess the performance of rotation estimation. The Open Source Computer Vision(OpenCV) algorithm was employed to recognize markers and measure their positions[72].

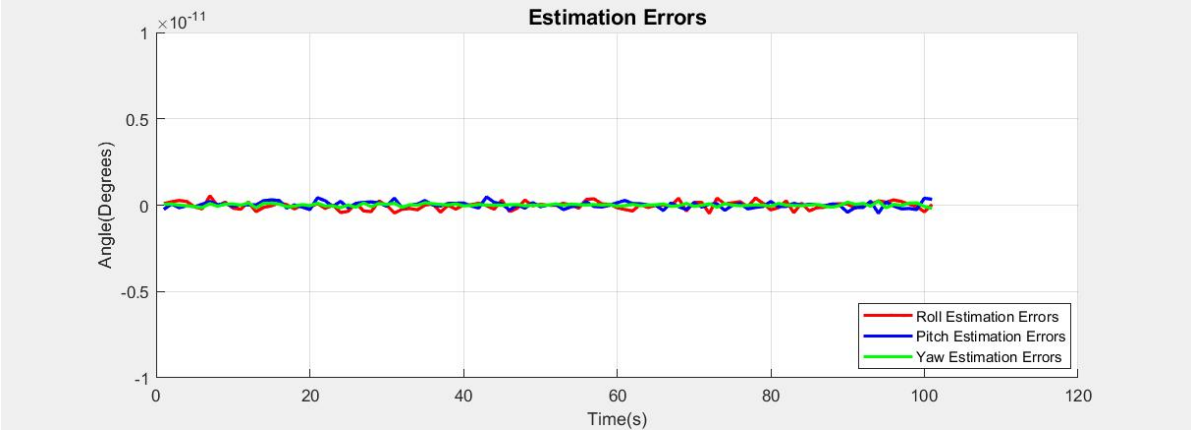
5.3.1 Simulation Model Test

Platform Model

The simulation model represents a navy ship flight deck equipped with nine markers: eight arranged in a circular pattern, and one positioned at the center of the platform. We assumed the quadrotor would hover at an altitude of 5 meters, tracking the center marker to maintain its position while the ship undergoes movement. In the simulation, the diameter of the marker circle is 1 meter, and the flight deck measures 10×10 meters. Two motion scenarios were tested: Case 1 involved rotation in one direction (Roll: -20° to 20°, Pitch: -10° to 10°, Yaw: -5° to 5°,), while Case 2 utilized a mathematical ship motion model (USS Oliver Hazard Perry class destroyer in Sea State 6)[73].



(a) Actual and Estimated Rotation Angles



(b) Estimation Error

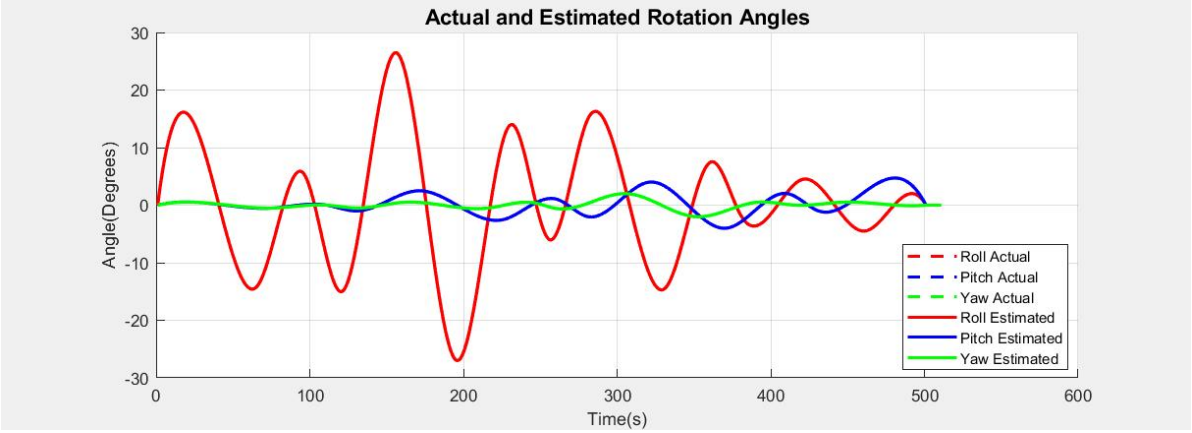
Figure 5.10. Case 1. Angles change in one direction. Actual angles are dashed lines, and estimated angles are solid lines.

Quadrotor Camera Model

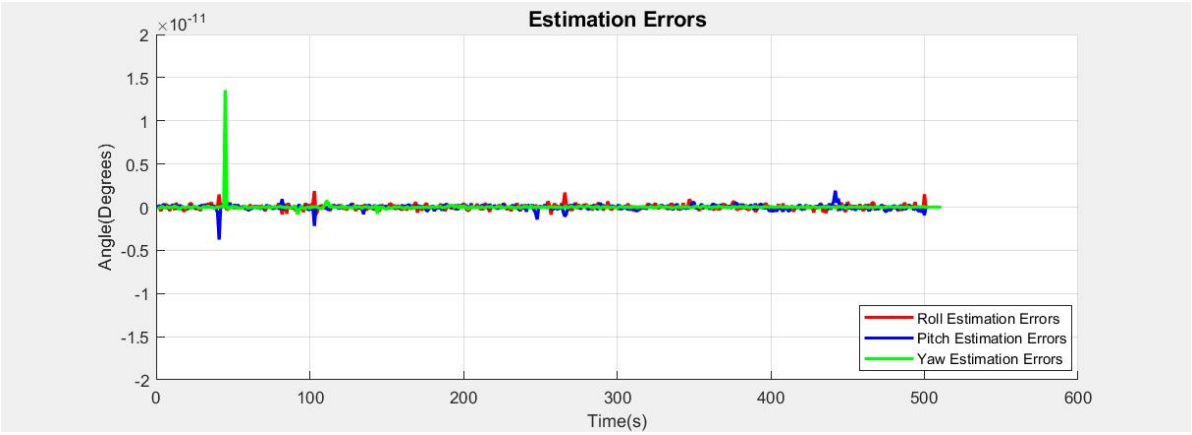
We assume that the quadrotor is equipped with the "Raspberry Pi Camera Module V2-8," which is equivalent to the hardware model. The camera has a pixel resolution of 3280×2464, and the focal angles are $45.4°$ horizontally and $31.1°$ vertically.

Result

The rotation estimation of each case is shown in Figures 5.10 and 5.11. Table 5.1 the Root Mean Square Errors(RMSEs) for each case and rotation. The RMSE for Case 1 ranged from $0.0771 \times 10^{-12°}$ to $0.2235 \times 10^{-12°}$, and for Case 2, it was between $0.3010 \times 10^{-12°}$ and



(a) Actual and Estimated Rotation Angles



(b) Estimation Error

Figure 5.11. Case 2. The ship motion model

$0.6083 \times 10^{-12^{\circ}}$. The results suggest that the estimation algorithm performs with minimal errors.

Table 5.1. Estimation RMSE (Degrees)

Motion	RMSE: Case 1	RMSE: Case 2
Roll	0.2235×10^{-12}	0.3010×10^{-12}
Pitch	0.1814×10^{-12}	0.3126×10^{-12}
Yaw	0.0771×10^{-12}	0.6083×10^{-12}

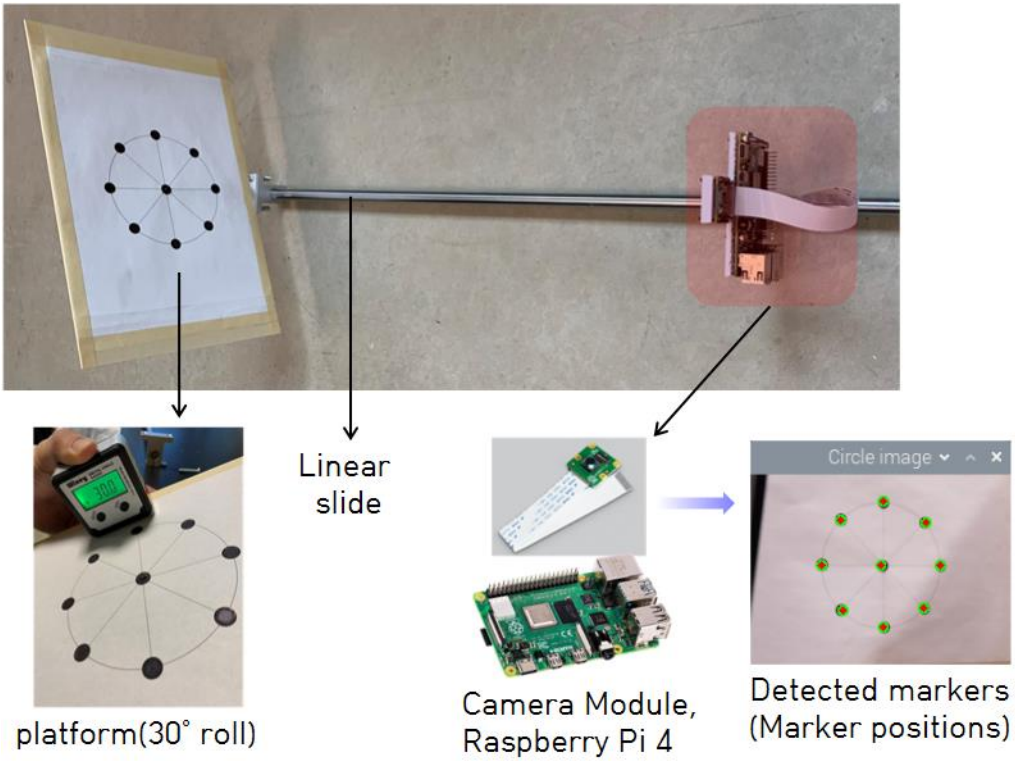


Figure 5.12. Hardware model image. The camera module, Raspberry Pi, and a plate with circular pattern markers are connected to a linear slide. The camera and the center mark are aligned. The camera detects markers, and the marker positions are obtained by the OpenCV algorithm.

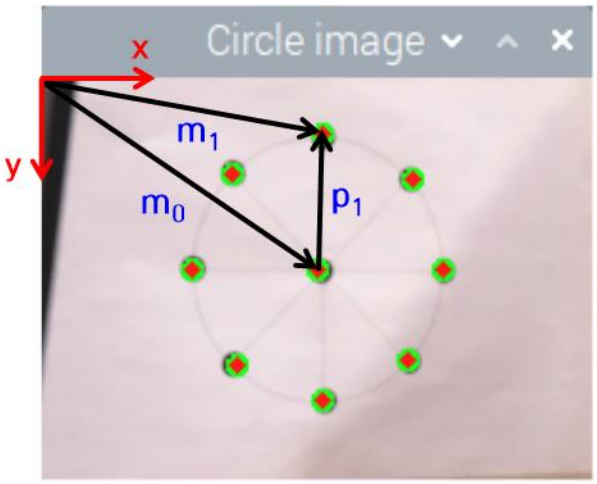


Figure 5.13. Detected circle image. m_0 is the center marker, m_1 is the bow marker position in the camera image coordinate.

5.3.2 Hardware Model Test

Figure 5.12 illustrates the hardware model used to test the rotation estimation algorithm. The hardware model's size was determined by scaling down the simulation model with a $1/10$ ratio. Thus, the distance between the camera center and the platform center was 50 centimeters, and the diameter of circular pattern markers was 10 centimeters. The platform mount was fixed at a 30-degree roll. In Figure 5.13, the i-th marker's 2D position from the center marker, $p_i = (X_{p_i}, Y_{p_i})$, can be calculated as follows,

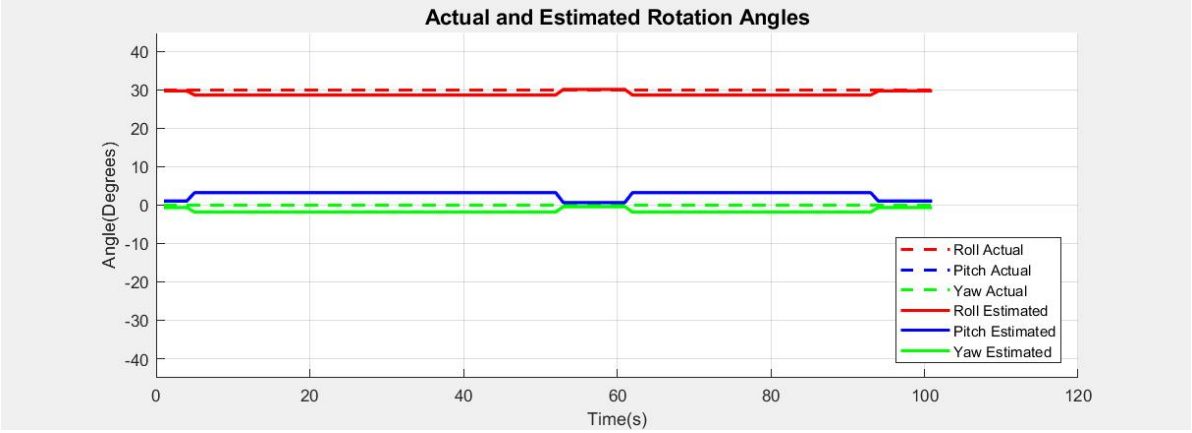
$$p_i = m_i - m_0.$$

The positions of markers are measured over 10 seconds with a 10Hz. Using marker positions p_i, the ellipse equation is determined by (5.3). Figure 5.14 displays the result of the hardware test, where the estimation errors tend to be consistent, and RMSEs were roll 1.1577°, pitch 2.9360°, and yaw 1.6080°.

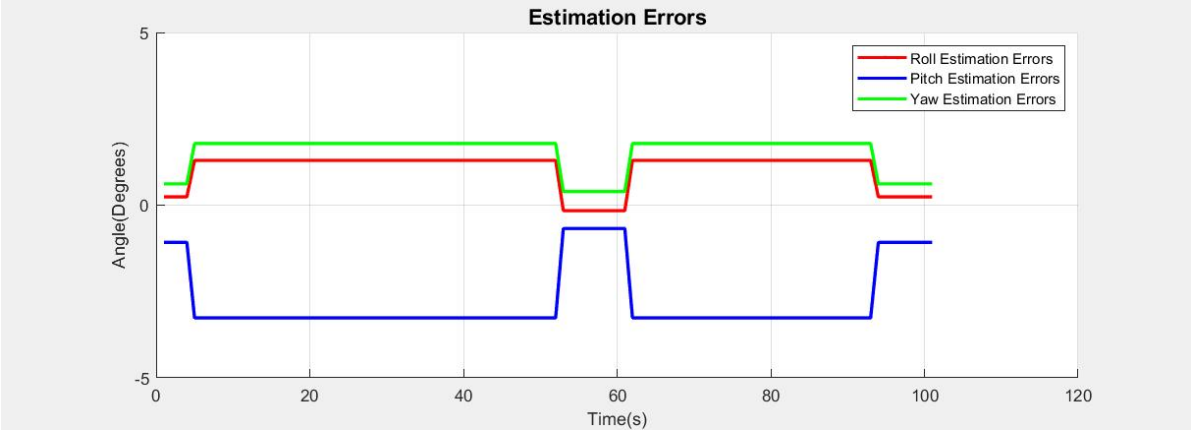
5.4 Analysis

This study introduces a rotation estimation method for a moving platform, specifically in the context of quadrotor landing. The proposed approach assumes the presence of a circular image with nine markers on the moving platform, and a quadrotor equipped with a camera tracks the platform's center. As the platform rotates, the circular marker images are distorted into ellipses on the camera's image plane due to the foreshortening effect. Analyzing these ellipses, determined by marker positions, allows for the estimation of the platform's rotation.

Both simulation and hardware models were tested, demonstrating that the rotation estimation algorithm achieved Root Mean Square Errors (RMSE) of less than $0.7 \times 10^{-12^\circ}$ degrees. Future work aims to enhance the algorithm and hardware model. Calibration of the hardware model will be pursued to minimize errors, considering the consistent estimation errors



(a) Actual and Estimated Rotation Angles



(b) Estimation Error

Figure 5.14. Hardware model experiment result

observed. Additionally, the study acknowledges the assumption of the quadrotor hovering precisely over the platform center. Future improvements will focus on extending the algorithm to estimate 6-DOF platform motions, including surge, sway, and heave.

To improve the hardware model, our plan involves utilizing a cable-driven motion simulator designed to replicate 6-DOF boat motions. The simulator will feature circular pattern markers on its platform, and a camera mounted on the simulator will be utilized to perform rotation estimation tests. This approach aims to refine and validate the proposed rotation estimation method under more realistic and dynamic conditions.

5.5 Acknowledgements

Support for this work was provided by Republic of Korea(ROK) Navy Sponsored Education Program.

Chapter 5, in full, is a reprint of the material as it appears in IEEE 8th International Conference on Control and Robotics Engineering(ICCRE), 2023, J. Jang, M. A. M. Macias and T. Bewley, “Camera Image Based Moving Platform Rotation Estimation for Quadrotor Landing”. The dissertation author was the primary investigator and author of this paper.

Chapter 6

A Novel Design of Pipe Inspection Robot

Pipe and duct networks constitute crucial components of global infrastructure, serving purposes such as water supply, natural gas transportation, waste management, and building ventilation[74]. However, the aging of these networks poses the risk of leaks, resulting in the loss of valuable resources and the release of hazardous substances[75][76]. Hence, regular inspection and maintenance are essential to minimize inefficiencies and safeguard the environment from potential harm. A significant challenge arises from the fact that a substantial portion of these networks is hidden underground, underwater, or within structures like buildings, refineries, ships, or submarines. Additionally, many older buried pipe systems lack accurate mapping, making it challenging to locate them accurately and increasing the risk of damage during nearby construction or repair projects involving extensive excavation. To address these challenges, the use of pipe inspection robots capable of internal inspection, exploration, and mapping becomes crucial. These robots are specifically designed to navigate complex pipe networks characterized by bends, joints, diameter variations, and vertical sections, ensuring their reliable maneuverability.

Numerous efforts have been made to develop efficient pipe inspection robots, with snake or serpentine-style robots representing common designs for this purpose[77][78][79]. However, these robots often exhibit characteristics of being lengthy, intricate, and having high blockage potential. Snake robots typically feature multiple treads or wheels around their circumference,

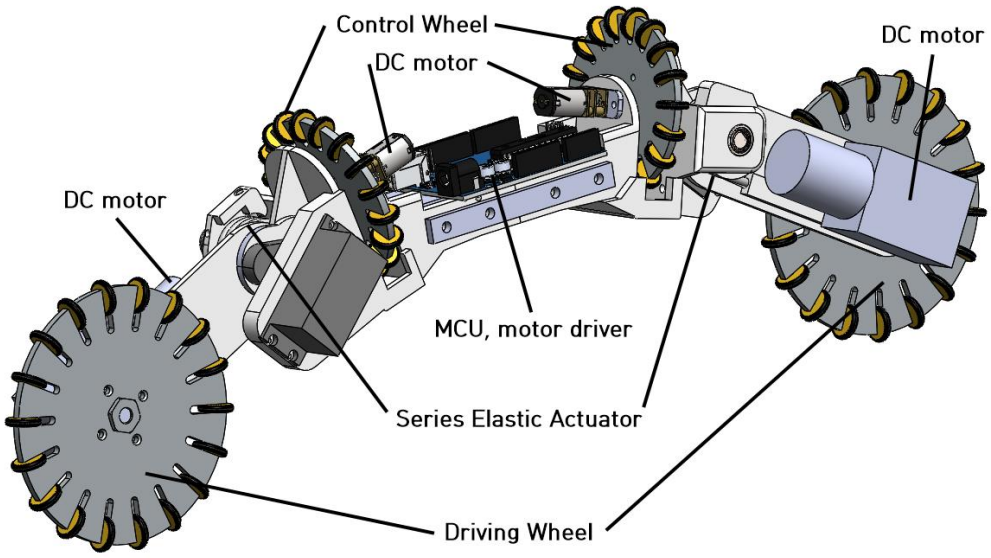


Figure 6.1. The overall configuration of the proposed pipe inspection robot Model S4.

which can extend outward to grip the pipe wall. These robots are frequently constructed with multiple independent modules, each equipped with various motors for articulation and locomotion. The Explorer II is an example of a pioneering snake robot design targeting a range of pipe diameters[80]. Another category of robots designed for pipe inspection is the screw drive in-pipe robot, which utilizes two large rotating screws for propulsion within the pipe[81][82]. However, these robots encounter challenges in navigating curves or joints. Some designs incorporate six or more wheels mounted on spring-loaded arms around the robot core [83]. Half of these arms are attached to one end of the vehicle, and the other half to the opposite end; each arm unfolds from the vehicle body as needed to grip the walls, enabling the robot to handle pipes with varying diameters. Despite their adaptability, these robots tend to be more complex and have higher blockage potential. Rover-type designs are also deployed in large mainline pipes, but they generally struggle to navigate into pipe branches at joints within pipe networks. These robots often feature one or more arms capable of pushing wheels against the opposite side of the pipe to enhance stability during rolling maneuvers[84][85].

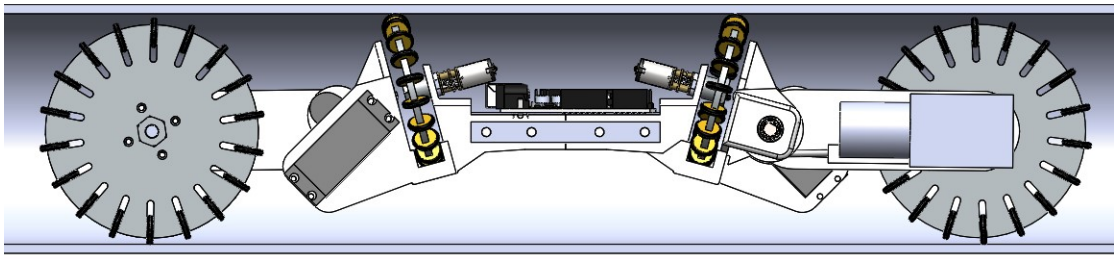
The robot proposed in this paper shares similarities with an existing tethered W-shaped design featuring pairs of omniwheels as drive wheels at each of the three interior joints and pairs

of half-spherical wheels at each end for control[86]. In comparison to this W-shaped design, known as AIRo-II, the proposed design, Model S4, is characterized by reduced complexity, fewer wheels, and lower blockage. Additionally, the AIRo-II design does not integrate Series Elastic Actuators(SEA) with springs to achieve inherent stability by passively adjusting the robot's width across the full pipe diameter range. The springs in AIRo-II firmly press the wheels against opposite walls at a minimum of three alternating contact points at all times. Notably, the roll control mechanism in the AIRo-II design involves two sets of spherical wheels, differing from the Model S4, which utilizes two orthogonally arranged omniwheels for driving purposes.

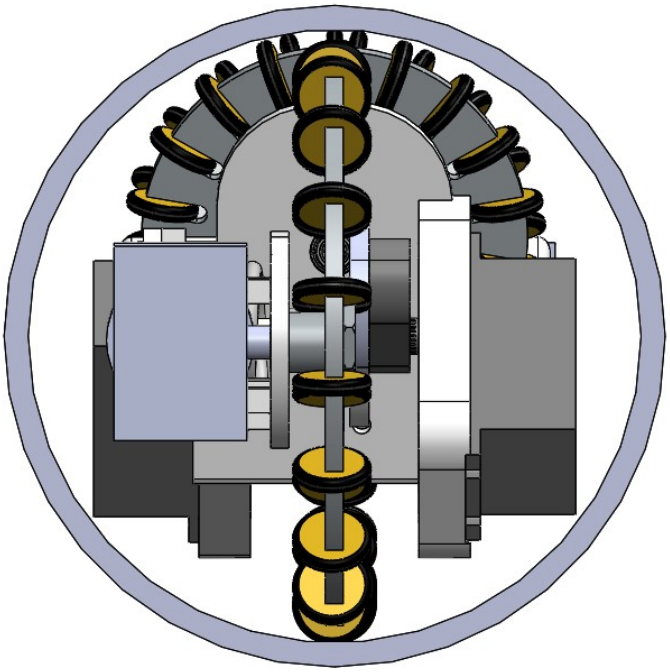
6.1 Design

6.1.1 Mechanical Design

The Model S4 is designed for pipe navigation, including traversing T-joints and curves. The overall configuration of Model S4 is illustrated in Figure 6.1, featuring two robot arms equipped with omniwheels driven by DC motors, referred to as Driving wheels, responsible for maneuvering the robot forward and backward. The robot arms are connected to a body frame through Series Elastic Actuators(SEAs), as detailed in 6.1.2. This SEA mechanical system generates continuous forces on the pipe wall to stabilize pitch and yaw motions, regardless of gravity's direction. This capability proves crucial when navigating through intricate pipe networks with curves, joints, and significant changes in diameter. Two omniwheels are affixed to the body frame, serving as Control wheels and powered by DC motors. These wheels have the capability to manage the robot's roll motion, given their vertical orientation to the pipe's axial center line. The primary operational objective of the model is to navigate effectively within a 4-inch pipe. All hardware components were chosen and designed to fit within the confines of a 4-inch diameter pipe, as depicted in Figure 6.2.



(a) Front-view



(b) End-view

Figure 6.2. Model S4 inside a 4-inch diameter pipe.

6.1.2 Series Elastic Actuator

DC motors or servo motors have become widely utilized in the field of robotics due to their recognized reliability, precise position control, and resistance to external disturbances. Their application has been extensively researched across various domains, including robot motion control, medical devices, compressors, and more[87]. However, these rigid actuators come with

inherent limitations when it comes to handling unexpected forces resulting from environmental impact or shock[88]. Exposure to significant forces can lead to damage to the shaft or gearboxes of these actuators. Moreover, for robots requiring the exertion of forces against the ground or a wall to maintain a stable pose, continuous actuation is required.

In contrast to stiff actuators, Series Elastic Actuators(SEAs) incorporate built-in elastic materials, specifically torsion springs. Unlike traditional stiff actuators that generate torques through electric current flow, SEAs derive torques from the deflection of an elastic material. While SEAs are recognized for their complex mechanisms, involving a motor, an elastic material, and configuration-dependent structures[89], they effectively reduce the stiffness associated with conventional actuators. This reduction enables SEAs to mitigate unintended forces directly on motor parts. Despite their complexity, SEAs have demonstrated their efficacy in applications such as robot legs, as exemplified by the Legged robot at MIT[90]. This characteristic allows SEAs to absorb unexpected shocks or disturbances resulting from impacts, enhancing overall stability and resilience[91]. The advantages offered by SEAs have led to their widespread adoption in wearable robots and exoskeleton devices[92]. As the torque is determined by the deflection of elastic materials, SEAs can also serve as force or torque sensors[93]. Additionally, the deflection of elastic materials in SEAs allows them to transmit force or torque to connected mechanisms without requiring additional motor actuation. Moreover, the inclusion of elastic materials, such as springs, in SEAs transforms the force control problem into a position control problem, leading to substantial improvements in force accuracy[94].

The research on Series Elastic Actuators (SEAs) primarily focuses on investigating the viability of applying pushing forces on a robot arm. The proposed SEAs aim to achieve inherent stability by passively expanding the robot across a wide range of narrow spaces, with the springs firmly pressing the wheels against opposite walls. Previous studies have explored various elastic components for SEAs, including compact and lightweight elastic beams[95], compression springs for linear motions[96], and torsion springs for rotational motions[97]. In this work, we propose the integration of torsion spring-motor in SEAs, as torque is necessary for continuously

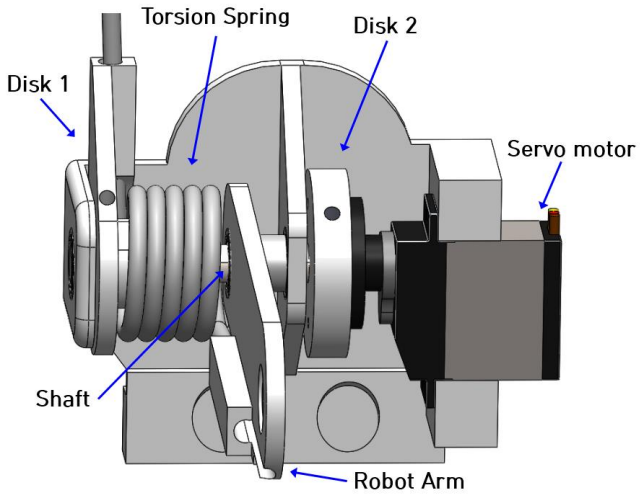


Figure 6.3. Type A SEA: Consists of a servo motor, a torsion spring, a D-profile shaft, 3D printed disks, and a robot arm.

pushing against the walls using the robot arm. The motor is affixed to the robot body frame or arm and is driven to deflect the torsion spring. As the torsion spring deflects, torque is applied to the robot arm connected to the spring. Two distinct types of torsion spring-motor integrated SEAs were designed and tested for characterization. Additionally, to address the limitations of these two types of SEAs, a conceptual design is suggested that can generate substantial torque using a DC motor, potentially finding applications in robots operating in diverse environments.

SEA Designs

This chapter explores two distinct types of Series Elastic Actuators(SEAs): Type A and Type B. These designs are specifically tailored for robots operating in confined spaces. The incorporation of a high-torque micro servo motor was chosen to streamline the design and enable precise control of the shaft angle. Although Type A and Type B SEAs share identical hardware components, their connection configurations are reversed.

Type A: Torsion Spring-Servo Integrated SEA

As depicted in Figure 6.3, Type A SEA incorporates a servo motor affixed to the body frame, with Disk 2 connected to the servo motor hub. A D-profile shaft is rigidly linked to both

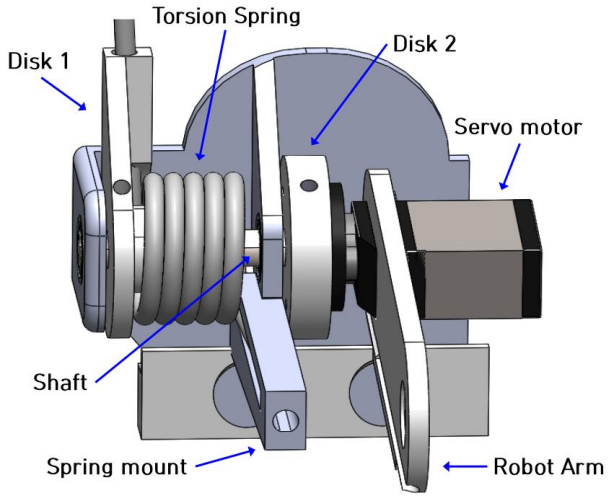


Figure 6.4. Type B SEA: Parts are connected in reverse order compared to the Type A SEA.

Disk 1 and 2. Disk 1 serves to connect the shaft and torsion spring, facilitating the transmission of motor torque to the spring. The torsion spring, firmly attached to a robot arm, bears the torque from the servo motor, inducing a change in the arm angle. However, the arm angle remains constant upon contact between the robot arm and the ground. Consequently, when additional torque is applied to the spring by the servo motor, the torsion spring initiates deflection. This design is particularly well-suited for constructing a robot arm capable of exerting continuous forces against walls or floors. The compact size of the Type A SEA proves advantageous for robots navigating narrow structures like pipes, enabling them to maintain stability and posture. Through the utilization of the Type A SEA, these robots can effectively apply forces to walls, ensuring their capability to navigate and operate within confined spaces.

Type B: Reverse Torsion Spring-Servo Integrated SEA

Figure 6.4 illustrates the configuration of the Type B SEA, a variant of the Type A SEA that shares identical hardware components. However, the arrangement of these components is reversed compared to the Type A SEA. In this design, a servo motor is integrated into a robot arm, with the servo motor hub firmly attached to Disk 2, and a D-profile shaft linking Disk 1 and Disk 2. A torsion spring connects Disk 1 to a spring mount on the body frame, and the deflection

of the torsion spring generates a pushing force on the robot arm. Notably, Type B lacks a servo motor mount on the body frame and has a smaller body width(Type A: 3.07”, Type B: 2.65”). This design offers advantages in fitting within smaller spaces.

SEA Experiment Setup

The hardware details for the experiment are outlined in TABLE 6.1. The SEA consists of a motor, a spring, and a robot arm connected in series. The objective of the experiment is to measure the forces exerted by the proposed SEA designs for characterization.

Table 6.1. Hardware Parts

Category	Specification
Motor (Stall torque)	RS100 Micro Servo (1.7037 N-m at 6V)
Shaft	6mm D-profile
Torsion Spring	Music wire
Microprocessor	Arduino Uno
Force sensor	Load cell force sensor

Torsion Spring Torque

The torque(τ) generated by a torsion spring can be computed based on Hooke's law[98],

$$\tau = k\triangle\theta, \tag{6.1}$$

where k is the spring rate, $\triangle\theta$ is the deflection angle of the torsion spring. The torsion spring rate k varies based on the spring configuration and is determined by[99],

$$k = \frac{Ed^4}{64DN_a} \text{ (N-m/rad)}. \tag{6.2}$$

In this equation, E denotes the modulus of elasticity, d represents the wire diameter, D is the mean diameter, and N_a is the number of active coils. The number of active coils(N_a) is defined

by the following equation, incorporating the spring coil number(N) and the leg length(L)[100],

$$N_a = N + \frac{L}{3\pi d}.$$

Table 6.2. Torsion Spring Specification

Category	Specification
E (Modulus of elasticity, N/m^2)	2.0684×10^{11}
d (Wire diameter, m)	0.0029
D (Mean diameter, m)	0.0219
N (Number of coils)	5.25
L (Leg length, m)	0.1016

Utilizing the torsion spring specifications outlined in Table 6.2, the spring rate of the torsion spring is determined to be 1.2005 N-m/rad. Concurrently, the measured torques are acquired by measuring forces at the edge of the robot arms and multiplying them by the length between the SEA shaft and the shaft hole on the arm.

Figure 6.5 depicts the side view of Type A SEA, and the torque generated by the SEA can be calculated by multiplying the force f by the horizontal distance r. Subsequently, the measured torque τ_m is expressed as,

$$\tau_m = rf = a\cos\phi f, \tag{6.3}$$

where ϕ is the initial arm angle measured by a protractor and is assumed not to be changing. The initial angle represents the angle at which the robot arm makes contact with the ground, marking the onset of spring deflection.

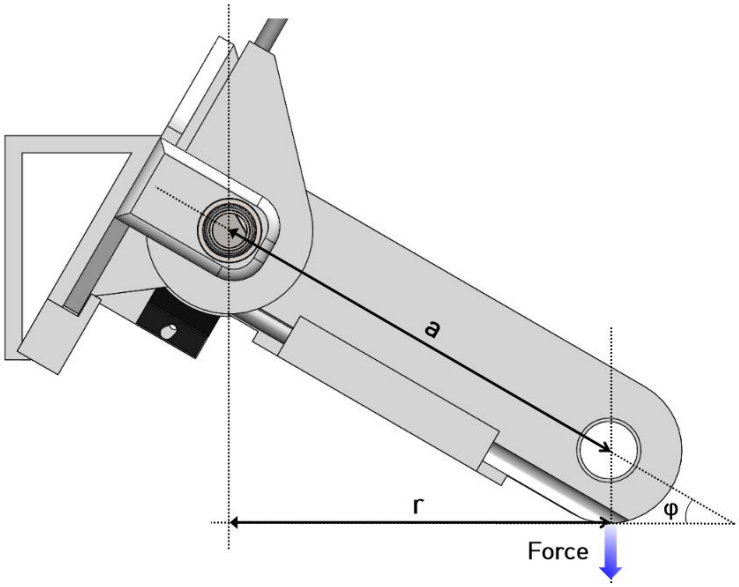
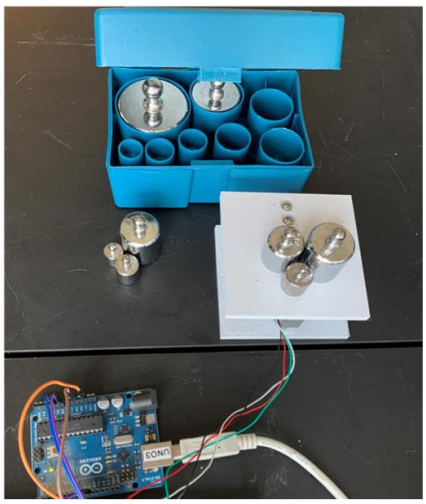


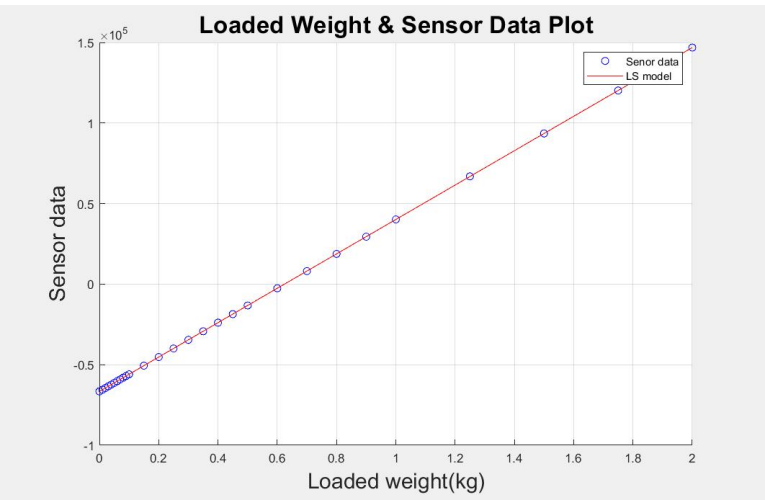
Figure 6.5. Type A SEA Side View: a is the arm length, and ϕ is the initial arm angle. r can be calculated as $r = a\cos\phi$. As the motor is actuated and the torsion spring is deflected, the force at the edge of a robot arm pushes against the ground.

(a) Force Sensor **(b)** Load Cell

Figure 6.6. Customized force sensor utilizing a load cell.

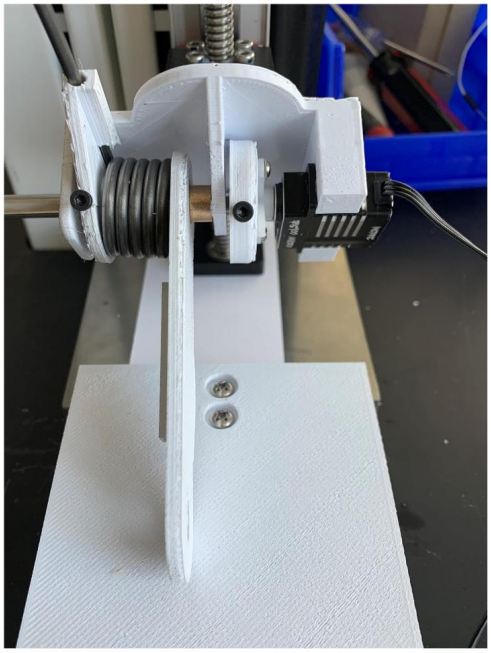


(a) Calibration

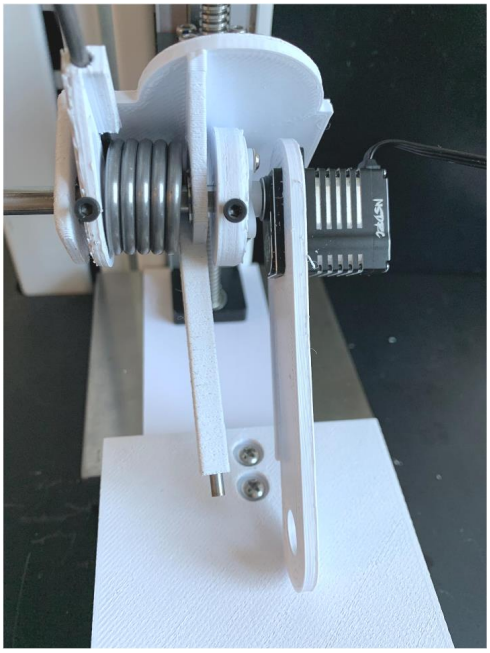


(b) Force sensor calibration plot

Figure 6.7. Calibrating the customized force sensor. Certain weights were added to the force sensor to obtain raw data. The plot shows the relationship between the loaded weight and the sensor's raw data. Blue circles are sensor data, and the solid red line is the LS model.



(a) Type A

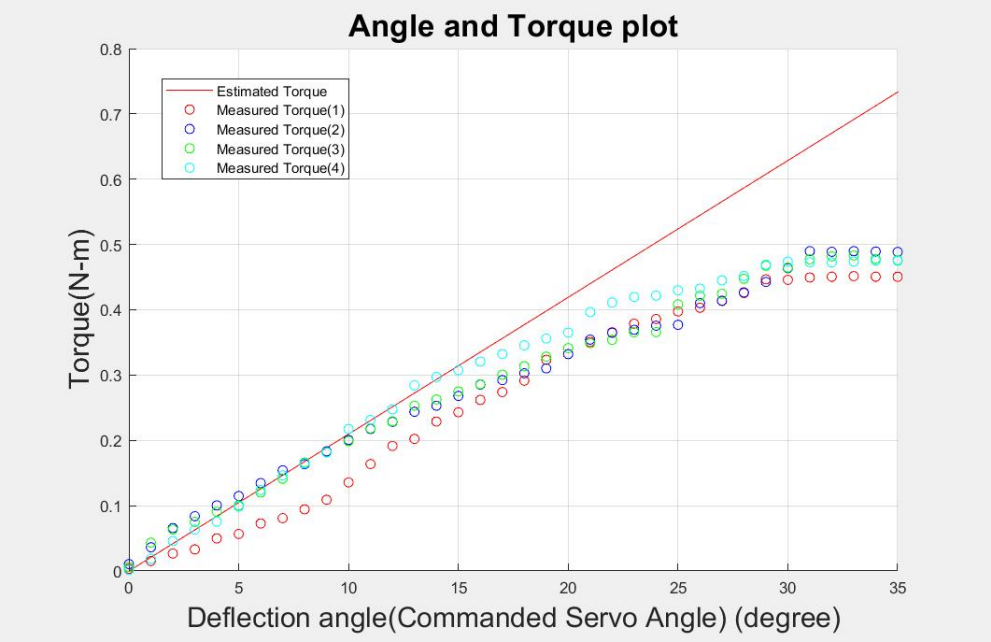


(b) Type B

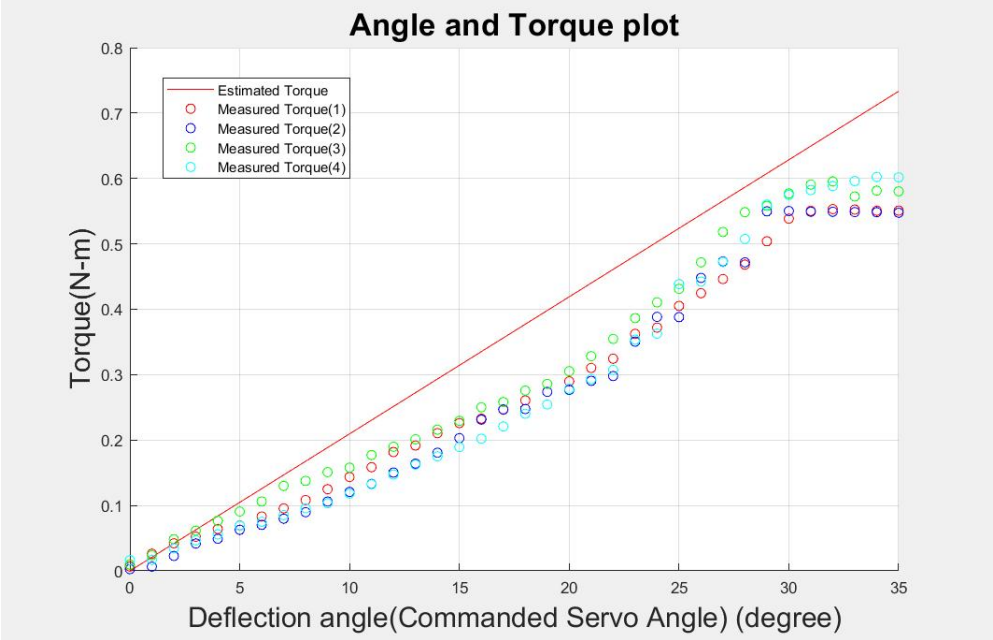
Figure 6.8. Type A and B Hardware Model.

Force Sensor

The torsion spring deflection results in a pushing force on the robot arm, as explained in 6.1.2. Instead of directly measuring the spring deflection angle, a force sensor can be employed



(a) Type A



(b) Type B

Figure 6.9. The relation between torsion spring deflection angle and the output torque of each SEA type. The deflection angle changes from 0 to 35 degrees, and four tests were performed for each model. Circles are torque(τ_m) calculated by (6.3) about commanded servo motor torque, and the solid red line is estimated torque(τ) computed by (6.1) about deflection angle.

to calculate the measured torque using (6.3). As illustrated in Figure 6.6a, a load cell force sensor was positioned beneath the robot arm to measure the force. This custom force sensor incorporates a load cell, as depicted in Figure 6.6b.

Calibration, as seen in Figure 6.7a, was conducted by adding specific weights to the force sensor. Raw data from the sensor were collected, and the Least Square(LS) model can be determined though (4.1) from (4.4). The calculated parameters were $a = 1.0675 \times 10^5, b = -0.6662 \times 10^5$. Figure 6.7b displays the sensor's raw data and LS model. The RMSE of the LS model compared to the measured weight calculated from the sensor's raw data was 2.0640×10^{-4} kg, and the sensor's raw data demonstrates a nearly linear relationship with the loaded weight.

SEA Experimental Results

Each SEA hardware test model was constructed, as depicted in Figure 6.8. The experiment aimed to measure the generated torque of proposed SEAs and analyze the error between the measured and estimated torque. The torque plots in Figure 6.9 illustrate the relationship between the deflection angle and torque. The measured torques of Type A and Type B in Figure 6.9a and 6.9b demonstrate a nearly linear relation from 0 to 30 degrees of deflection angle, but it remains constant beyond 30 degrees. This observation can be attributed to the limited torque provided by the servo motor, which is insufficient to alter the angle of the torsion spring beyond a certain angle(30 degrees). Considering the data from 0 to 30 degrees deflection angle, the RMSEs and NRMSEs for estimated torques for each type of SEA are as follows: Type A: 0.0797 N-m (12.68%) and Type B: 0.0897 N-m (14.28%). Note that the difference between the maximum and minimum values of the estimated torques was used as a scaler to normalize RMSE to calculate NRMSE. The average RMSE and NRMSE of both designs were 0.0847 N-m (13.48%).

Possible Design for High-torque Requirement Robots

In the results, it was observed that Type A and Type B SEAs had a limitation of a 30-degree deflection angle due to the stall torque of the servo motor. To address the need for

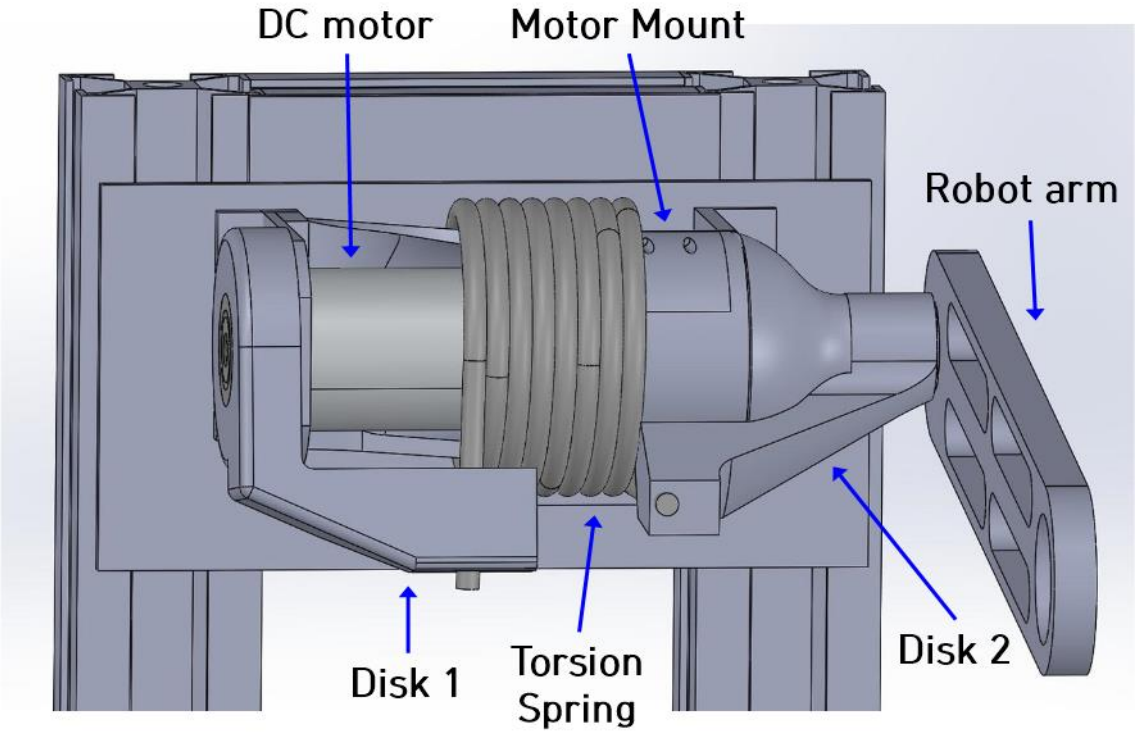


Figure 6.10. Type C SEA: Consists of a high-torque DC motor, a customized torsion spring, 3D printed disks, and a robot arm.

higher torque requirements, Type C was designed, as seen in Figure 6.10. Type C SEA is a larger size designed explicitly for high-torque-required robots. It incorporates a high-gear ratio DC motor with an encoder to generate the necessary torque. Type C SEA shares the same connection sequence of hardware parts as Type A SEA. However, the high-torque DC motor used in Type C allows for greater deflection of the torsion spring. The motor is positioned inside the torsion spring on the body frame to maintain a low-profile body width. The motor shaft is connected to Disk 1, which is linked to the torsion spring, and the other side of the torsion spring is connected to Disk 2. When the motor is actuated, the torsion starts to deflect, generating torque that applies a force to the robot arm connected to Disk 2. Considering the larger diameter of the torsion spring in Type C, selecting spring properties such as wire diameter and the number of coils is crucial to achieving the required spring rate defined as (6.2).

SEA Test Results Analysis

This SEA research introduces torsion spring-servo motor integrated SEAs designed to generate continuous forces on a robot arm. These designs leverage the torsion spring's capability to generate torques through deflection, eliminating the need for continuous motor actuation. The experimental results show that the output torques of both designs exhibit a nearly proportional to the deflection angle. However, it was observed that there are deflection angle limits due to the stall torque of the servo motor used in the experiment. The Type C design is suggested to overcome this limitation, which utilizes a high-torque DC motor for greater deflection capabilities, suitable for scenarios requiring higher torque output.

Future work will focus on enhancing the hardware to minimize potential sources of errors, particularly related to the use of low-cost 3D printed parts like the disks. These parts may introduce slip and result in torque loss from the servo motor. To address this, it is necessary to replace these 3D-printed components with more robust materials, such as metal, to enhance reliability. Additionally, further tests and characterization of the Type C design are needed, involving the construction of hardware and conducting tests. In conclusion, this paper proposes various types of SEAs with unique characteristics, offering flexible solutions for accommodating diverse space constraints and torque demands in robotic applications.

6.1.3 Hardware

Model S4's prototype was constructed utilizing a 3D-printed body frame crafted from Formlabs' Rigid 4000 resin, renowned for its durability, heat resistance, and stability [**?**]. This model integrates Series Elastic Actuators(SEAs) consisting of a torsion spring, a high-torque, low-profile servo motor, a D-profile metal shaft, 3D-printed and laser-cut D-profile Disks, and a robot arm. The servo motor boasts a maximum torque capacity of 3922.66mNm, adequate for managing the deflection and loading of the torsion spring, reaching a maximum torque of 3163.58mNm. The Driving wheel DC motors are worm gear motors with a rated torque of 980.67mNm,

while the Control wheel DC motors are geared motors with a rated torque of 196.13mNm. All motors are under the control of an Arduino Uno microprocessor and TB6612FNG dual motor driver. The robot is equipped with two different diameters of omniwheels: 100mm (3.93 inches) diameter for Driving wheels and 70mm (2.76 inches) diameter for Control wheels. Their centers precisely aligned with the pipe's axial plane. The DC motors are responsible for actuating the omniwheels to manage forward, backward, and roll motions. The hardware components utilized in constructing Model S4 are detailed in Table 6.3.

Table 6.3. Model S4 Prototype Hardware

Hardware	Supplier	Model
DC motor(Driving)	Yosoo	12V 10RPM Worm gear motor
DC motor(Control)	Greartisan	12V 78RPM gear motor
Servo motor	Sincecam	40kg High Torque Servo
Omniwheel(Driving)	GTF Robots	100mm Aluminum Omniwheel
Omniwheel(Control)	GTF Robots	70mm Aluminum Omniwheel
Torsion spring	Mcmaster	9271K592
SEA Shaft	Robot shop	6mm D Shaft
Bearing	Mcmaster	57155K585, 7804K141
Microprocessor	Arduino	Uno
Motor controller	Pololu	TB6612FNG
Power Supply	AlloverPower	3-24V DC Power Supply Kit

* Body frame, Robot arm, SEA Disk 2 are 3D printed, and SEA Disk 1 is laser cut aluminum.

6.2 Static Analysis

Analyzing the statics of Model S4 offers valuable insights into the necessary forces required to maintain a stable pose and navigate within the pipe. This understanding is crucial for selecting hardware components, particularly motors, that can provide the required torques.

6.2.1 Driving Wheel and Servo Motor Torques

Figure 6.11 illustrates the free-body diagram of Model S4's front view within a tilted 4-inch pipe. The statics parameters are outlined in Table 6.4. r_d and r_c denote the radii of the

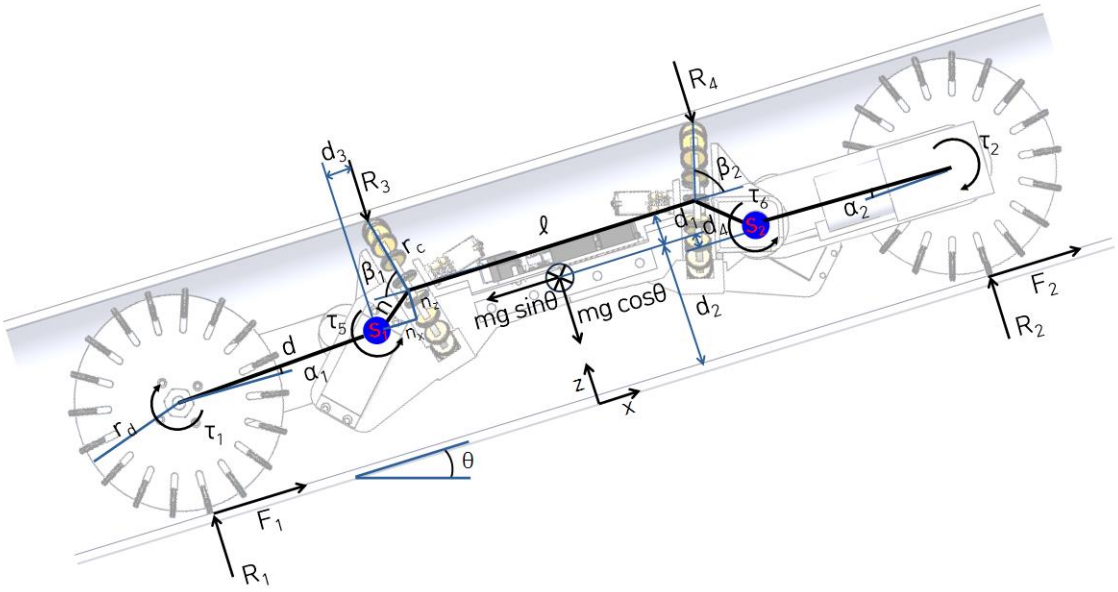


Figure 6.11. Statics(Front-view): Required Driving wheel and SEA torques for driving a robot and maintaining the stable pose inside the tilted pipe with pitch angle θ.

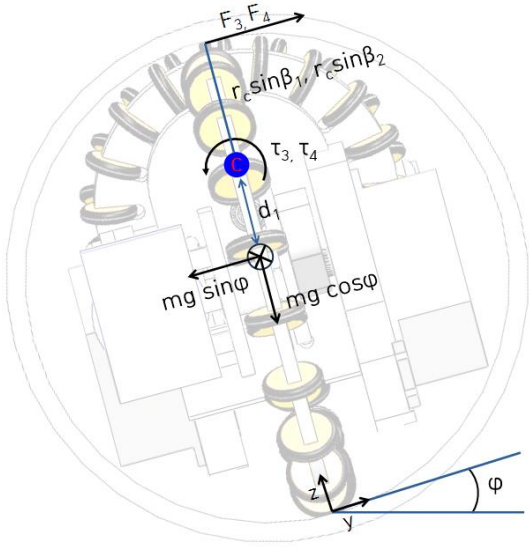


Figure 6.12. Statics(End-view): Required Control wheel torques for the roll control with roll angle ϕ.

Table 6.4. Statics Parameters

Categories	Parameters	Magnitude
Driving wheel radius	r_d	50.00mm
Control wheel radius	r_c	35.00mm
Arm length	d	92.18mm
Body frame length	l	135.50mm
SEA distance	$n(n_x, n_z)$	22.92mm, 16.09mm
Arm angle	α_1, α_2	0.63° (4 inch pipe)
Control wheel angle	β_1, β_2	75°
Total mass	m	950g
Distance	d_1, d_2, d_3, d_4	13.87mm, 53.93mm, 15.32mm, 2.96mm
Roll, Pitch, Yaw	ϕ, θ, ψ	-
Wheel force	F_1, F_2, F_3, F_4	-
Wheel torque	$\tau_1, \tau_2, \tau_3, \tau_4$	-
SEA torque	τ_5, τ_6	-
Reaction force	R_1, R_2, R_3, R_4	-

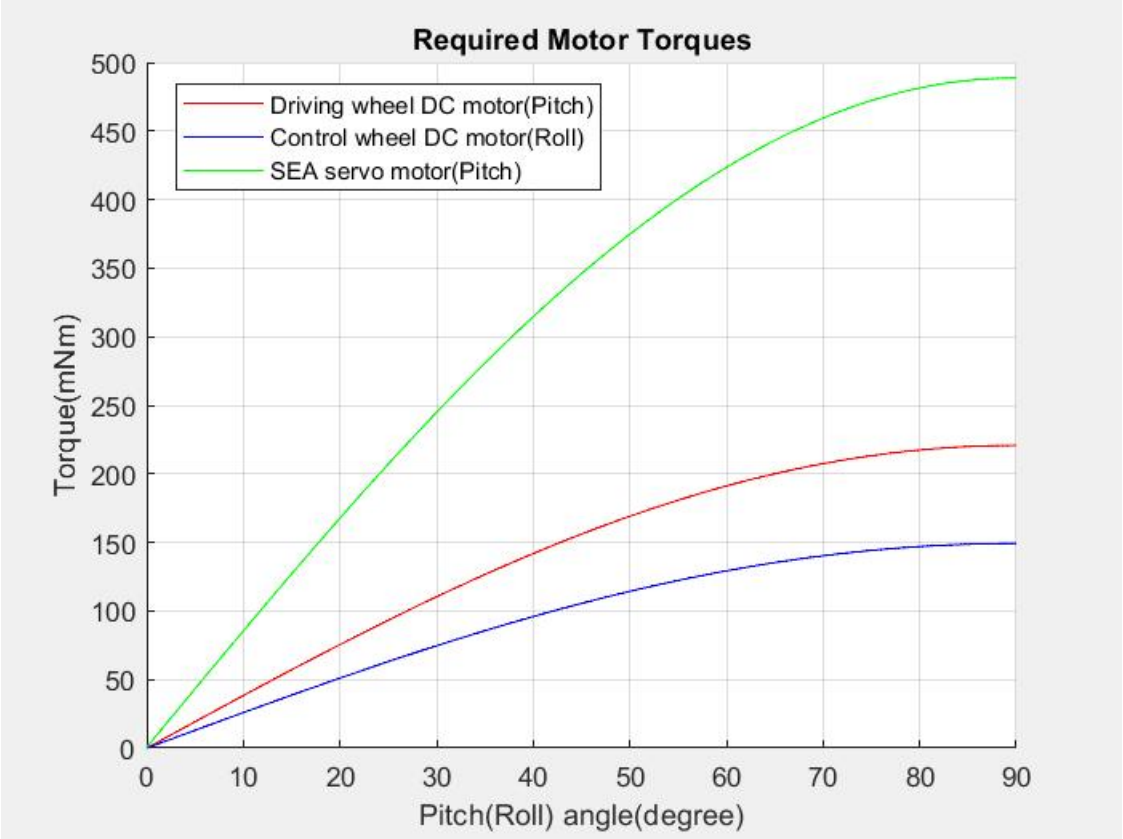


Figure 6.13. Required motor torques about pitch (roll) angle.

driving and control wheels, respectively. d represents the arm length from the SEA shaft center, and l is the body length, indicating the distance between the two Control wheel centers. n is the length between the Control wheel and SEA shaft centers. The angles α_1, α_2 denote the arm angles, while β_1, β_2 indicate the Control wheel angles. As the pipe tile angle is θ, gravitational forces can be expressed as $mg\sin\theta$ and $mg\cos\theta$ in the x and z axes. According to Newton-Euler equations, the sum of forces and torques at static equilibrium can be expressed as follows,

$$
\begin{aligned}
&\sum F_x : F_1 + F_2 = mg\sin\theta, \\
&\sum F_z : R_1 + R_2 = R_3 + R_4 + mg\cos\theta, \\
&\sum \tau_w : \tau_1 + \tau_2 = r_d F_1 + r_d F_2, \\
&\sum \tau_{s1} : \tau_5 = -F_1(d\sin\alpha_1 + r_d) - R_1(d\cos\alpha_1) - d_3 R_3 - (n_x + \frac{l}{2})\frac{mg\cos\theta}{2} + d_4\frac{mg\sin\theta}{2}, \\
&\sum \tau_{s2} : \tau_6 = -F_2(d\sin\alpha_2 + r_d) - R_2(d\cos\alpha_2) - d_3 R_4 - (n_x + \frac{l}{2})\frac{mg\cos\theta}{2} + d_4\frac{mg\sin\theta}{2}, \\
&\sum \tau_b : R_1(\frac{l}{2} + n_x + d\cos\alpha_1) = R_3(\frac{l}{2} + r_c\cos\beta_1) + d_2 F_1,
\end{aligned}
\tag{6.4}
$$

where F_1 and F_2 represent the forces propelling the robot forward or backward, and R_1, R_2, R_3, and R_4 denote the reaction forces on the wheels. τ_w and τ_s indicate the required torques of the Driving wheel DC motor and SEA servo motors. τ_b is the sum of torque at the mass center located in the middle of the body frame. Assuming symmetry between the left and right halves regarding control wheel and arm angles, forces, and torques, (6.4) can be simplified and solved for τ_w and τ_s,

$$
\begin{aligned}
&\tau_w = \frac{r_d mg\sin\theta}{2}, \\
&\tau_s = c_1 R_1 + d_3 R_3 + c_2 F + c_3\frac{mg\cos\theta}{2} - d_4\frac{mg\sin\theta}{2},
\end{aligned}
\tag{6.5}
$$

with constants defined as,

$$c_1 = d\cos\alpha, \quad c_2 = d\sin\alpha + r, \quad c_3 = n_x + \frac{l}{2},$$
$$c_4 = \frac{l}{2} + n_x + d\cos\alpha, \quad c_5 = \frac{l}{2} + r\cos\beta.$$

Furthermore, the reaction forces can be expressed as,

$$R_1 = \frac{c_5 mg\cos\theta - 2d_2F}{2(c_5 - c_4)}, \quad R_3 = \frac{c_4R_1 - d_2F}{c_5}.$$

Figure 6.13 illustrates the necessary motor torques computed by (6.5). The required DC motor torques, denoted as τ_w, vary with the pipe tilt angle θ. The maximum torque needed for climbing the vertical pipe was 220.73mNm. The variable τ_s undergoes changes as the pipe tilt angle varies. Within the 4-inch pipe, the arm angles are constant at 0.63°. Hence, τ_s is a function of θ, and Figure 6.13 represents the required SEA torque to maintain the robot pose inside the pipe depending on the pipe tilt angle. The maximum τ_s occurs when the pipe is vertical, amounting to 488.66mNm.

6.2.2 Control Wheel Torque

Figure 6.12 is the free body diagram of the end-view. As the robot rolls with an angle ϕ, the sum of forces and the torque on the Control wheel center can be written as,

$$\sum F_y : F_3 + F_4 = mg\sin\phi, \tag{6.6}$$
$$\sum \tau_c : \tau_3 + \tau_4 = F_3 r_c \sin\beta_1 + F_4 r_c \sin\beta_2. \tag{6.7}$$

where d_1 is the distance between the Control wheel center and the mass center. τ_c represents the required torque of the Control wheel motor. Assuming that both the Control wheels' forces and

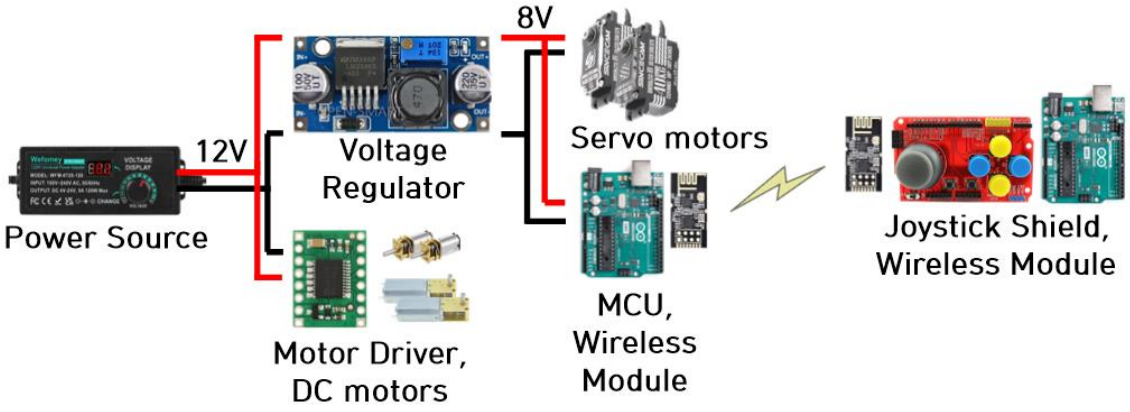


Figure 6.14. Electonics setup for Model S4 control.

torques are the same, then (6.7) can be rewritten about τ_c,

$$\tau_c = \frac{(r_c \sin\beta) mg \sin\phi}{2}. \tag{6.8}$$

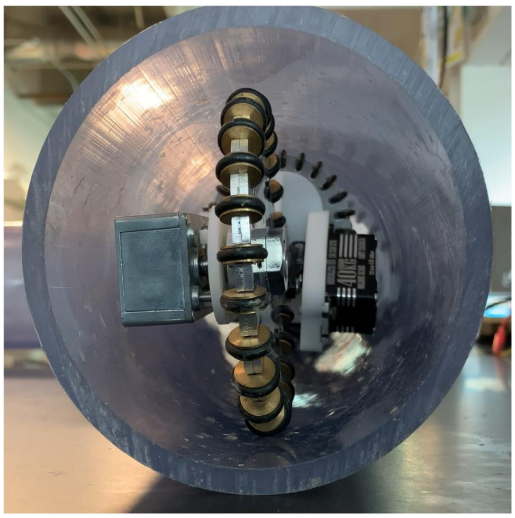
As expressed in (6.8), the roll angle ϕ influences the required torque, as depicted in Figure 6.13. The maximum torque reached 149.24mNm when the roll angle was 90$\circ$.

6.3 Experiment

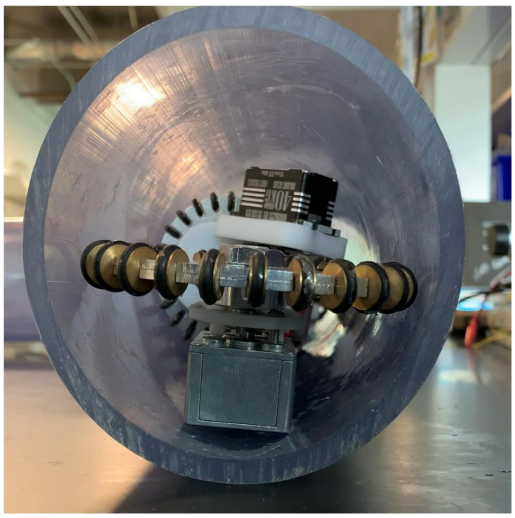
The prototype Model S4 is operated by a user, and the electronics setup is illustrated in Figure 6.14. The power source supplies 12 volts to the motor driver(TB6612FNG) and DC motors. A voltage regulator(LM2596) converts power to 8 volts for servo motors and a microprocessor. A wireless module(NRF24L01) enables wireless communication with the Arduino joystick shield. The joystick is utilized to control forward-backward motions, roll, and arm angles.

6.3.1 Roll Motion

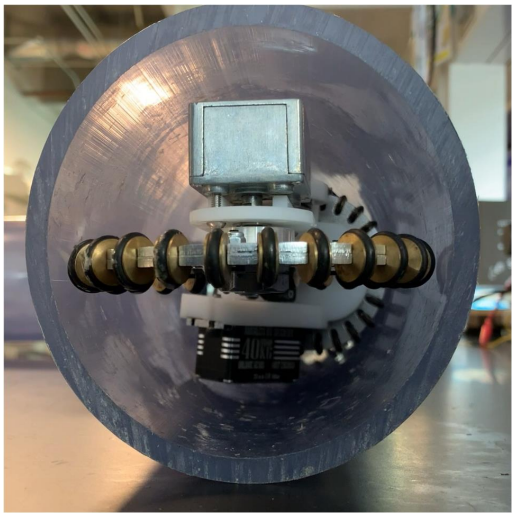
In a roll control test conducted inside the 4-inch diameter horizontal pipe, the robot demonstrated its ability to roll by actuating the control wheel motors, which are orthogonally installed to the body and driven by geared motors. As indicated in 6.2, each motor's required



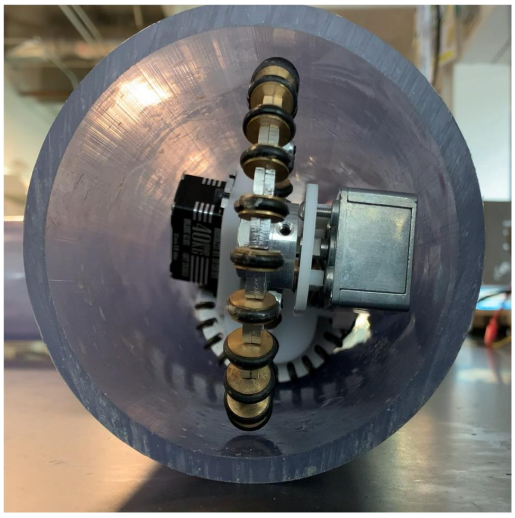
(a) 0° Roll.



(b) 90° Counter clock wise.



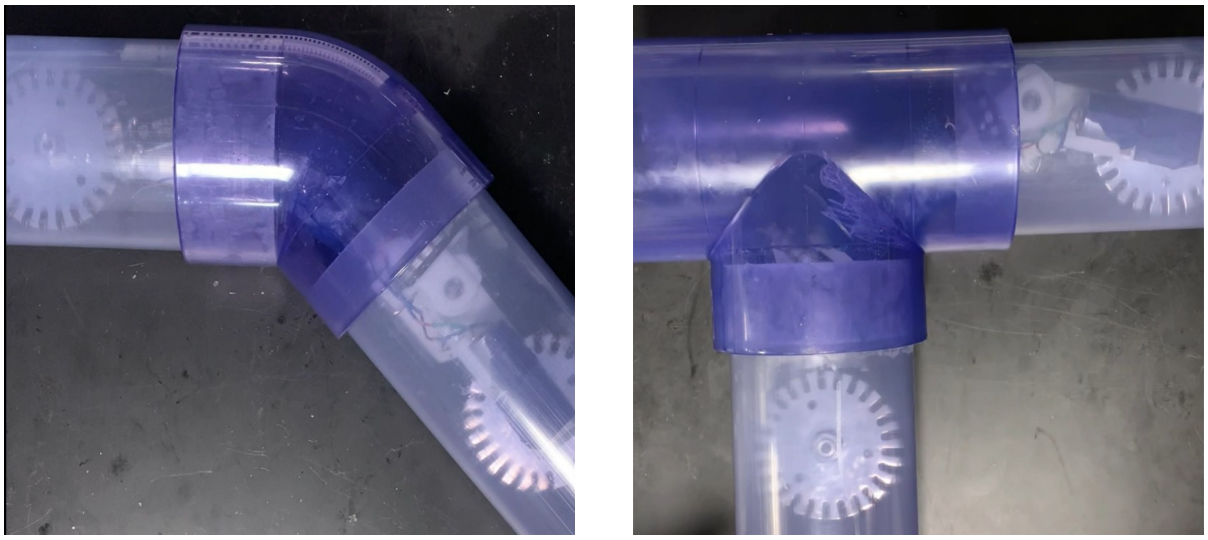
(c) 90° Clock wise.



(d) 180° Roll.

Figure 6.15. Roll motion test inside the 4-inch pipe.

torque for roll control reaches a maximum of 149.24mNm at a 90° roll angle. The control wheel motors, with a rated torque of 193.13mNm, provide sufficient torque to control the robot's roll at any angle. Figure 6.15 depicts the roll motion test inside the pipe, showcasing the robot's capability to roll in both directions up to 90° and 180° angles. The Series Elastic Actuators(SEAs) generate continuous pushing forces against the wall, ensuring the robot maintains stability during the roll test. The video of the roll test experiment can be viewed at the following link: https://youtu.be/sGsYZAuReSo



(a) Curve **(b)** T-joint

Figure 6.16. Model S4 inside the curve and T-joint.

6.3.2 Curve Maneuver

In the pipe networks, which include both angled curves and straight pipes, Figure 6.16a illustrates the robot navigating inside a 4-inch diameter curve. Upon reaching the curve, the robot continues its forward movement, simultaneously adjusting the rear robot arm to press against the pipe wall by modifying the rear arm angle. As the robot negotiates the corner of the curve, the Control wheels are activated to induce a roll until the robot's pose aligns vertically. This strategic approach prevents the robot from becoming stuck inside the corner, ensuring a smooth traversal around the curve and allowing it to proceed to the next section of the pipeline. The video demonstrating the curve maneuver test experiment is available at the following link: https://youtu.be/W5P9tWS4jkk

6.3.3 T-joint Maneuver

Figure 6.16b showcases the robot navigating inside a 4-inch diameter T-joint. The strategy employed for T-joint maneuvering is depicted in Figure 6.17. Leveraging the continuous grabbing forces provided by the SEAs of Model S4 to the robot arms against the pipe wall, a

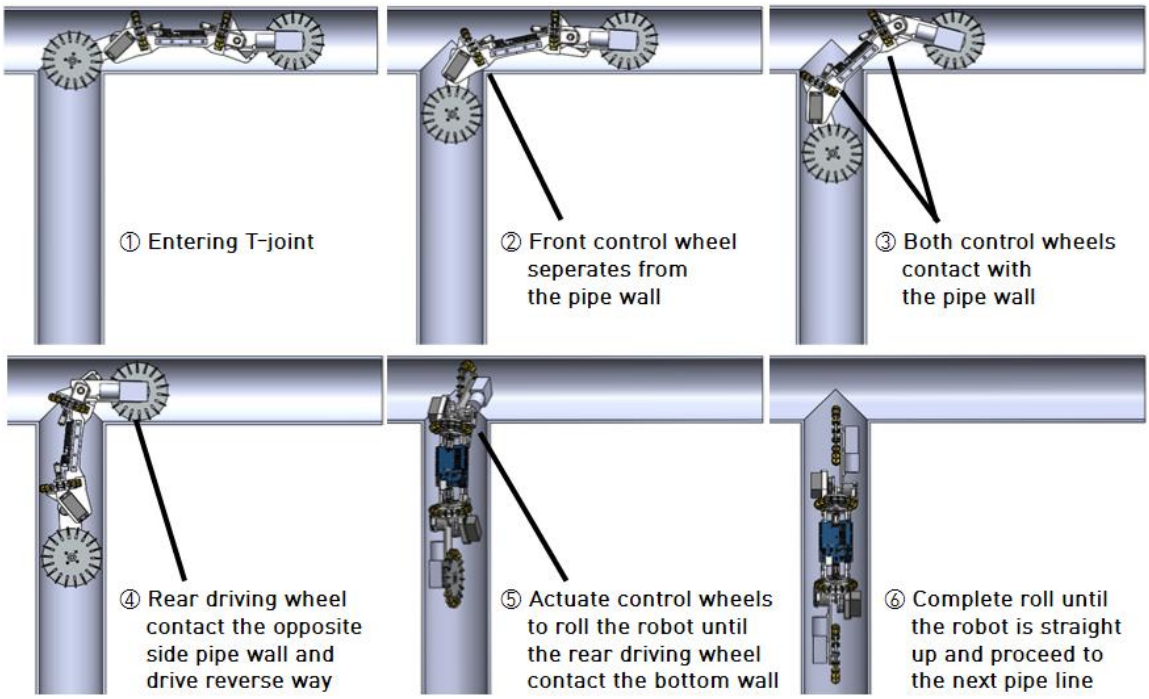


Figure 6.17. 4-inch diameter T-joint pipe maneuvering.

minimum of three wheels are in constant contact with the wall during the traversal of a T-joint.

Upon entering the T-joint, the front Driving wheel moves into the joint, causing the front Control wheel to separate from the pipe wall. Arm angles are then adjusted by actuating SEA servo motors, ensuring the maintenance of grabbing forces and a stable pose until the front Control wheel attaches to the next pipeline wall. Subsequently, by modifying the arm angle, the rear driving wheel comes into contact with the opposite side pipe wall. As the rear driving wheel makes contact with the pipe wall, the rear driving wheel is actuated in the reverse direction, providing the necessary pushing force for the robot to move forward. During the continued maneuvering of the robot to the next pipeline, the rear driving wheel experiences slippage at the corner of the T-joint. To address this slippage, the Control wheels are actuated to roll the robot until the rear driving wheel contacts the bottom pipe wall. Simultaneously, the Driving wheels are actuated forward to complete the roll until the robot is in an upright position, enabling it to proceed to the next pipeline. The T-joint experiment video is available at the following link: https://youtu.be/WpIB3E1lksE

6.4 Analysis

In this chapter, we introduced a novel pipe inspection robot design named Model S4, designed specifically for operations within a 4-inch pipe network. The robot is equipped with four omniwheels driven by DC motors, with two wheels controlling forward and backward motions and the other two aligned orthogonally to the body centerline for roll control. Additionally, a torsion spring and a servo motor embedded in a Series Elastic Actuator(SEA) enable the adjustment of robot arm angles, providing continuous force against the pipe wall, particularly during maneuvers such as turning curves or navigating T-joints. The static analysis conducted provides the necessary torques for both DC motors and SEA servo motors, which is crucial for the selection of appropriate hardware components. Tests were carried out to validate the mobility of the proposed mechanical design within pipeline curves and joints. The results demonstrate the robot's capability to navigate through curves and T-joints, showcasing its potential for exploring complex pipeline environments. Future work will concentrate on the development of an autonomous navigation control algorithm tailored for pipeline operations. A feedback control system will play a pivotal role in achieving autonomous navigation, utilizing an Inertial Measurement Unit (IMU) sensor to maintain a stable pose during operations, especially when traversing curves and T-joints. Given the reliance on battery power, the algorithm will prioritize energy-efficient operation, ensuring thorough exploration of all pipelines with minimal travel distance. Additionally, the incorporation of camera vision for pipe damage detection will be explored and implemented in future iterations of the robot.

6.5 Acknowledgements

Support for this work was provided by Republic of Korea(ROK) Navy Sponsored Education Program.

Chapter 6, in part, has been submitted for publication of the material as it may appears in IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS) 2024, J. Jang,

Z. Hooker, T. Wan, P. Tejeda, et al., “Design and Characterization of the Torsion Spring-Motor Integrated Series Elastic Actuator”. Also, in part, has been submitted for publication of the material as it may appears in IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS) 2024, J. Jang, T. Wan, Z. Hooker, P. Tejeda, T. Bewley, “A Novel Design of 4-Omniwheel Robot for Interior Exploration and Inspection of Pipe Network”. The dissertation author was the primary investigator and author of both papers.

Chapter 7

Conclusion

This thesis discusses three main topics. First, the cable-driven boat motion simulator is a mechanical system that has a platform connected to eight cable moves replicating the 6-DOF boat motion. Since cable-driven parallel robots(CDPRs) use cables that are only capable of pulling an object but cannot push it, it is crucial to maintain the proper cable tensions. In either case, when cable tensions are too high or too low, the system can fail, the structure can be damaged due to very tight cables, or they cannot move properly as the cables are slack. Controllers of cable-driven parallel robots are required to control the motions of the platform and perform cable tension optimization. Our eight cable-6 DOF simulator is an underdetermined system that has many tension solutions at the equilibrium. We suggested a Control Lyapunov Function(CLF)-Quadratic Program(QP) controller to control motion and optimize cable tensions. The simulation and hardware model experiments allowed us to validate and refine our approach to controlling the boat motion simulator. A simulation model was suggested to test the performance of the controller. 6-DOF periodic motion and ship model motion were used as desired motions. The results illustrate that the CLF-QP controller can be useful to control the platform motion of CDPRs with tension limits set by users. The CLF-QP controller controls a platform motion with small errors and maintains cable tensions in the desired boundaries without optimization violations. The hardware model test focused on applying the optimal tension calculated by the CLF-QP controller and testing the motion control accuracy by leveraging a motion capture

system. The test results reveal that the desired tension, computed using CLF-QP methods with CDPR dynamics, effectively propels the platform towards the desired state, particularly evident in the low-frequency test. However, we have also observed many tasks to improve the results in hardware parts and algorithms. In addition to the control problem, workspace analysis for parameter optimization was considered. This research examines workspace analysis to optimize the parameters of the simulator. The study considers three parameters: platform height, width, and cable attachment point, to investigate their impact on the feasible workspace. The results indicate that increasing platform height leads to a smaller workspace when the platform rolls but a larger workspace when a roll is not applied. Platform width significantly affects the workspace, with smaller platforms allowing for a larger feasible workspace in all cases. Also, when the cable attachment point is small and close to the platform's corners, the workspace becomes smaller due to the high chance of cable-to-cable interference. A larger cable attachment point usually results in a larger workspace because the distance between cables is farther. Overall, the findings from simulation tests suggest that smaller platform widths and larger cable attachment points are beneficial for achieving a larger workspace. However, it is essential to consider practical constraints such as platform size, which provides enough space for electronics and sensors on the platform. Also, there should be sufficient room to perform landing tests for specific applications, such as developing an automatic aerial vehicle landing on a moving platform.

The Second topic was camera image-based rotation estimation. This study presents the moving platform rotation estimation algorithm for the landing of aerial vehicles, especially considering a quadrotor equipped with a camera. Assuming that a circular image with nine markers is on the moving platform, and a quadrotor tracks the platform center. A quadrotor camera obtains a circular image on its image plane. Camera images are distorted because of the foreshortening effect. Thus, if the platform is rotated, circular marker images become ellipses on the image plane. By analyzing the ellipse determined by marker positions, the rotation estimation of a platform can be achieved. Simulation and hardware models were tested, and the results showed that the rotation estimation algorithm was performed with less than 0.710^{-12} degrees

RMSE. We suggested a platform pose estimation algorithm, leveraging 2D circular patterns camera images. The validation through hardware model tests lays a foundation for more accurate and reliable algorithms.

Finally, we introduced an innovative pipe inspection robot design, denoted as Model S4, designed specifically for operation within 4-inch pipes. This robot incorporates four omniwheels, each powered by DC motors. Two of these wheels facilitate the robot's forward and backward movements, while the other two are oriented orthogonally to the body's centerline to enable effective roll control. Notably, the robot integrates a Series Elastic Actuator (SEA) featuring a torsion spring and a servo motor. This SEA mechanism allows for dynamic adjustments in the robot's arm angles, ensuring a continuous force against the pipe wall during operations, particularly when navigating curves or T-joints. Through static analysis, we gained valuable insights into the necessary torques for both the DC motors and SEA servo motors, aiding in the selection of appropriate hardware components. Rigorous tests were conducted to validate the mobility of the proposed mechanical design, demonstrating the robot's capability to navigate through pipeline curves and joints successfully. The results affirm the effectiveness of the Model S4 in exploring intricate pipeline configurations.

Chapter 8

Future Work

8.1 Cable-driven Boat Motion Simulator

Future work will focus on enhancing hardware capabilities to facilitate the development of a real-time feedback control system. This involves incorporating state estimation techniques such as Kalman filtering and forward kinematics for improved accuracy. Additionally, there is a plan to build a larger model to extend the scope of testing, particularly for drone landing scenarios. The final goal of the cable-driven boat motion simulator project is to build a large-size model to develop the landing control system by using actual drones. To achieve this objective, the simulator must be built in scale-up size to actually test the landing of the aerial vehicle used in maritime operations. A more precise hardware system is required for a larger simulator. Based on the experience of the small-size hardware test, the cable level winding is needed to avoid cable entanglement.

8.2 Camera Image-based Platform State Estimation

The future agenda for Camera Image-based Rotation Estimation involves refining the algorithm and advanced hardware tests. Since estimation errors were consistent, a hardware model could be calibrated by performing further tests to minimize errors. In addition, the OpenCV algorithm needs improvement for more accurate circle detection. In our study, a quadrotor is assumed to be hovering on the platform center. However, as the platform moves

in 6-DOF motions, there must be a tracking error causing the drones not to be aligned to the platform center. Thus, the algorithm needs to have the capability to estimate not only the rotation of the platform but also the translational motions, including surge, sway, and heave. Additionally, in the hardware test, the calibration has to be done with the stationary platform to correct the consistent estimation errors, and the further test has to be done with a dynamically moving platform. Thus, estimation equations need to be modified to account for the offset in the ellipse center position. Advanced hardware tests will be performed using a cable-driven motion simulator. Circular pattern markers will be set up on the simulator's platform, and a camera will be located on the simulator to perform the state estimation for the dynamically moving platform.

8.3 Pipe network Inspection Robot

For the Pipe Network Inspection Robot, our future plans involve the integration of sensors to build a feedback control system for autonomous maneuvering. This includes the incorporation of IMU sensors. The IMU sensor provides roll angle data, facilitating the actuation of control wheels for effective roll control. Utilizing the driving wheel motor encoder data and IMU sensor data will enable precise control over forward and backward motions. Furthermore, an encoder sensor embedded in the Series Elastic Actuator(SEA) will measure the robot arm angle, allowing for the calculation of the force generated by the SEA and control over the arm angle. The ongoing work primarily focuses on modifying the robot design to accommodate these sensors, involving the redesign of the body frame to incorporate an encoder sensor and a disk for arm angle measurement. Subsequent testing will evaluate the functionality and efficacy of these sensors. Simultaneously, our efforts will extend to the development of an automated navigating system algorithm, aiming to optimize the robot's path for efficient exploration of pipe networks, especially by saving the battery of the automated version of the robot. These future initiatives are geared towards enhancing the capabilities and reliability of each system, opening paths for broader applications, and improving overall performance.

Bibliography

[1] B. Dasgupta, T.S. Mruthyunjaya, "The Stewart platform manipulator: a review," Mechanism and Machine Theory, Volume 35, Issue 1, pp. 15-40, ISSN 0094-114X, 2000.

[2] K. Talke, D. Drotman, N. Stroumtsos, M. de Oliveira, and T. Bewley, "Design and Parameter Optimization of a 3-PSR Parallel Mechanism for Replicating Wave and Boat Motion," In 2019 International Conference on Robotics and Automation (ICRA), IEEE Press, pp. 7955–7961, 2019.

[3] P. Miermeister et al., "The CableRobot simulator large scale motion platform based on cable robot technology," 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea (South), pp. 3024-3029, 2016.

[4] R. Skelton and M. de Oliveira, "Tensegrity systems," Springer, pp. 17-19, 2019.

[5] T. Bewley, "Stabilization of low-altitude balloon systems, Part 2: riggings with multiple taut ground tethers, analyzed as tensegrity systems," University of California, San Diego, Unpublished, 2019.

[6] J. Jung, J. Piao, S. Park, J. Park and S. Y., "Ko. 2016. Analysis of cable tension of high speed parallel cable robot: High speed position tracking of winch," 2016 16th International Conference on Control, Automation and Systems (ICCAS), pp. 1053-1056, 2016.

[7] A. Ghasemi, M. Eghtesad and M. Farid, "Workspace analysis of redundant cable robots", 2008 World Automation Congress, pp. 1-6, 2008.

[8] X. Diao and O. Ma, "Workspace Analysis of a 6-DOF Cable Robot for Hardware-in-the-Loop Dynamic Simulation," 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4103-4108, 2006.

[9] D. Q. Nguyen, and M. Gouttefarde, "On the Improvement of Cable Collision Detection Algorithms," Mechanisms and Machine Science, pp. 29–40, 2015.

[10] R. Mersi, S. Vali, M. S. haghighi, G. Abbasnejad and M. T. Masouleh, "Design and Control of a Suspended Cable-Driven Parallel Robot with Four Cables," 2018 6th RSI International Conference on Robotics and Mechatronics (IcRoM), Tehran, Iran, pp. 470-475, 2018.

[11] A. Z. Loloei, M. M. Aref and H. D. Taghirad, "Wrench feasible workspace analysis of cable-driven parallel manipulators using LMI approach," 2009 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Singapore, pp. 1034-1039, 2009.

[12] P. Gallina, A. Rossi, and R. Williams, "Planar Cable-Direct-Driven Robots, Part II: Dynamics and Control," Proceedings of the ASME Design Engineering Technical Conference, 2001.

[13] M. Zarebidoki, A. Lotfavar, and H. Fahham, "Dynamic Modeling and Adaptive Control of a Cable-suspended Robot," Proceedings of the World Congress on Engineering 2011, WCE 2011, pp. 2469-2473, 2011.

[14] A. Aflakiyan, H. Bayani, and M. T. Masouleh, "Computed torque control of a cable suspended parallel robot," 2015 3rd RSI International Conference on Robotics and Mechatronics (ICROM), Tehran, pp. 749-754, 2015.

[15] T. Bewley, "Numerical renaissance," Renaissance Press, 2017.

[16] A. Pott, T. Bruckmann, and L. Mikelsons, "Closed-form Force Distribution for Parallel Wire Robots," Fraunhofer IPA, 2009.

[17] J. Jang, and T. Bewley, "Tension optimization of the 6-DOF cable-driven boat motion simulator," RSAE 2021, Paris, France, 2021.

[18] Y. He, and J. Han, "Control Lyapunov Functions: New Framework for Nonlinear Controller Design", IFAC Proceedings Volumes, Volume 41, Issue 2, pp. 14138-14143, 2008.

[19] Y. Wang and S. Boyd, "Fast Evaluation of Quadratic Control Lyapunov Policy, IEEE Transactions on Control Systems Technology", vol. 19, no. 4, pp. 939-946, July 2011.

[20] K. Galloway, K. Sreenath, A. D. Ames and J. W. Grizzle, "Torque Saturation in Bipedal Robotic Walking Through Control Lyapunov Function-Based Quadratic Programs," in IEEE Access, vol. 3, pp. 323-332, 2015.

[21] R. Barbieri, "Lynx helicopter landing on ship in rough sea [Video]," YouTube, https://www.youtube.com/watch?v=NJIZTL2ZyEw, 2013.

[22] W. Chinsatitf and T. Saitoh, "Improvement of eye detection performance for inside-out camera," 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), Okayama, Japan, pp. 1-6, 2016.

[23] X. Ying and H. Zha, "Geometric Interpretations of the Relation between the Image of the Absolute Conic and Sphere Images," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 12, pp. 2031-2036, 2006.

[24] Q. Nguyen and K. Sreenath, "L1 adaptive control for bipedal robots with control Lyapunov function based quadratic programs," 2015 American Control Conference (ACC), Chicago, IL, USA, 2015, pp. 862-867, doi: 10.1109/ACC.2015.7170842.

[25] M. A. Martinez, "USV Attitude Position Estimation by a Hovering UAV Using Monocular Images of DeckMounted Lights," UC San Diego, 2022.

[26] A. Pott and V. Schmidt, "On the forward kinematics of cable-driven parallel robots," 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 2015, pp. 3182-3187, doi: 10.1109/IROS.2015.7353818.

[27] J. Begey, L. Cuvillon, M. Lesellier, M. Gouttefarde, and J. Gangloff, "Dynamic Control of Parallel Robots Driven by Flexible Cables and Actuated by Position-Controlled Winches," in IEEE Transactions on Robotics, vol. 35, no. 1, pp.286-293, Feb. 2019, doi: 10.1109/TRO.2018.2875415.

[28] R. E. Skelton, R. Adhikari, J. . -P. Pinaud, Waileung Chan and J. W. Helton, "An introduction to the mechanics of tensegrity structures," Proceedings of the 40th IEEE Conference on Decision and Control (Cat. No.01CH37228), Orlando, FL, USA, 2001, pp. 4254-4259 vol.5, doi: 10.1109/CDC.2001.980861.

[29] R. Skelton and M. de Oliveira, "Tensegrity systems," Springer, pp. 17-19, 2019.

[30] T. Bewley, "Stabilization of low-altitude balloon systems, Part 2: riggings with multiple taut ground tethers, analyzed as tensegrity systems," University of California, San Diego, 2019, Unpublished.

[31] Y. Yang, and J. Lee, "Design of a Control Lyapunov Function for Stabilizing Specified States," IFAC Proceedings Volumes. 43, pp.529-534, 2010, doi: 10.3182/20100705-3-BE-2011.00088.

[32] Hassan K. Khalil, "Nonlinear Control global edition," Pearson, 2015.

[33] J. Choi, F. Castañeda, C. Tomlin, and K. Sreenath, "Reinforcement Learning for Safety-Critical Control under Model Uncertainty, using Control Lyapunov Functions and Control Barrier Functions," 2020, doi: 10.15607/RSS.2020.XVI.088.

[34] J.L. Sanchez-Lopez, J. Pestana, S. Saripalli, and P. Campoy, "An Approach Toward Visual Autonomous Ship Board Landing of a VTOL UAV," J. Intell. Robotics Syst. 74, 1–2, April 2014, pp. 113–127, doi: https://doi.org/10.1007/s10846-013-9926-3

[35] M. M. Aref and H. D. Taghirad, "Geometrical workspace analysis of a cable-driven redundant parallel manipulator: KNTU CDRPM," 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France, 2008, pp. 1958-1963, doi: 10.1109/IROS.2008.4650670.

[36] S. Chen and J. Feng, "Research on detection of fabric defects based on singular value decomposition," The 2010 IEEE International Conference on Information and Automation, Harbin, China, 2010, pp. 857-860, doi: 10.1109/ICINFA.2010.5512449.

[37] J. Yang, S. Wongsa, V. Kadirkamanathan, S. A. Billings and P. C. Wright, "Metabolic Flux Estimation-A Self-Adaptive Evolutionary Algorithm with Singular Value Decomposition," in IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 4, no. 1, pp. 126-138, Jan.-March 2007, doi: 10.1109/TCBB.2007.1032.

[38] M. Fabritius, G. Rubio-Gómez, C. Martin, J. C. Santos, W. Kraus and A Pott, "A nullspace-based force correction method to improve the dynamic performance of cable-driven parallel robots," Mechanism and Machine Theory, Volume 181, 2023, 105177, ISSN 0094-114X, https://doi.org/10.1016/j.mechmachtheory.2022.105177.

[39] W. B. Lim, S. H. Yeo and G. Yang, "Optimization of Tension Distribution for Cable-Driven Manipulators Using Tension-Level Index," in IEEE/ASME Transactions on Mechatronics, vol. 19, no. 2, pp. 676-683, April 2014, doi: 10.1109/TMECH.2013.2253789.

[40] M. A. Carpio-Alemán, R. J. Saltaren-Pasmino, A. Rodriguez, G. A. Portilla, J. S. Cely, N. X. Gonzalez-Alvarez, and J. M. Castillo-Guerrero, "Collision and Tension Analysis of Cable-Driven Parallel Robot for Positioning and Orientation," 2018 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC), Ixtapa, Mexico, 2018, pp. 1-6, doi: 10.1109/ROPEC.2018.8661464.

[41] M. Grant, and S. Boyd, CVX: Matlab software for disciplined convex programming, version 2.0 beta. http://cvxr.com/cvx, September 2013.

[42] M. Grant, and S. Boyd, Graph implementations for nonsmooth convex programs, Recent Advances in Learning and Control (a tribute to M. Vidyasagar), V. Blondel, S. Boyd, and H. Kimura, editors, pp. 95-110, Lecture Notes in Control and Information Sciences, Springer, 2008.

[43] B. Lee, V. Saj, and M. Benedict, "Machine Learning Vision and Nonlinear Control Approach for Autonomous Ship Landing of Vertical Flight Aircraft," The Vertical Flight Society, 2021.

[44] J. L. Sanchez-Lopez, J. Pestana, S. Saripalli, and P. Campoy, "An Approach Toward Visual Autonomous Ship Board Landing of a VTOL UAV," J. Intell. Robotics Syst. 74, 1–2 (April 2014), pp. 113–127, doi: https://doi.org/10.1007/s10846-013-9926-3.

[45] S. Jaiteh, S. F. Adillah Suhaimi, T. Ching Seong, A. M. Buhari, L. Lini and H. Neyaz, "Smart Scale Tracking System Using Calibrated Load Cells," 2019 IEEE Conference on Sustainable Utilization and Development in Engineering and Technologies (CSUDET), Penang, Malaysia, 2019, pp. 170-174, doi: 10.1109/CSUDET47057.2019.9214692.

[46] Nasir K., Shauri R. L. A. and Jaafar J., "Calibration of embedded force sensor for robotic hand manipulation," 2016 IEEE 12th International Colloquium on Signal Processing & Its Applications (CSPA), Melaka, Malaysia, 2016, pp. 351-355, doi: 10.1109/CSPA.2016.7515859.

[47] M. Sharifzadeh, M. T. Masouleh and A. Kalhor, "Design, construction & calibration of a novel Human-Robot Interaction 3-DOF force sensor," 2015 3rd RSI International Conference on Robotics and Mechatronics (ICROM), Tehran, Iran, 2015, pp. 182-187, doi: 10.1109/ICRoM.2015.7367781.

[48] T. Nakano, K. Nagata, M. Yamada and K. Magatani, "Application of least square method for muscular strength estimation in hand motion recognition using surface EMG," 2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Minneapolis, MN, USA, 2009, pp. 2655-2658, doi: 10.1109/IEMBS.2009.5332858.

[49] Y. Xu, K. Xu, J. Wan, Z. Xiong and Y. Li, "Research on Particle Filter Tracking Method Based on Kalman Filter," 2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), Xi'an, China, 2018, pp. 1564-1568, doi: 10.1109/IMCEC.2018.8469578.

[50] G. Welch and G. Bishop, "An Introduction to the Kalman Filter," Technical Report, University of North Carolina at Chapel Hill, USA, 1995.

[51] K. A. Talke, "Hanging tether management for unmanned air - surface vehicle teams," (Order No. 28718622), UC San Diego; ProQuest Dissertations & Theses A&I, 2021, (2584293491). Retrieved from https://www.proquest.com/dissertations-theses/hanging-tether-management-unmanned-air-surface/docview/2584293491/se-2

[52] R. Barbieri, "Lynx helicopter landing on ship in rough sea [Video]," YouTube, 2013 January, https://www.youtube.com/watch?v=NJIZTL2ZyEw

[53] R. Liu et al., "Vision-guided autonomous landing of multirotor UAV on fixed landing marker," 2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), Dalian, China, 2020, pp. 455-458, doi: 10.1109/ICAICA50127.2020.9182512.

[54] J. Kim, S. Woo and J. Kim, "Lidar-guided autonomous landing of an aerial vehicle on a ground vehicle," 2017 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), Jeju, Korea (South), 2017, pp. 228-231, doi: 10.1109/URAI.2017.7992719.

[55] P. Wu, Z. Shi and P. Yan, "Improved EKF-SLAM Algorithm of Unmanned Helicopter Autonomous Landing on Ship," 2018 37th Chinese Control Conference (CCC), Wuhan, China, 2018, pp. 5287-5292, doi: 10.23919/ChiCC.2018.8483338.

[56] J. L. Sanchez-Lopez, J. Pestana, S. Saripalli, and P. Campoy, "An Approach Toward Visual Autonomous Ship Board Landing of a VTOL UAV," 2014, J. Intell. Robotics Syst, 74, 1–2 (April 2014), pp. 113–127, https://doi.org/10.1007/s10846-013-9926-3.

[57] M. F. Sani and G. Karimian, "Automatic navigation and landing of an indoor AR. drone quadrotor using ArUco marker and inertial sensors," 2017 International Conference on Computer and Drone Applications (IConDA), Kuching, Malaysia, 2017, pp. 102-107, doi: 10.1109/ICONDA.2017.8270408.

[58] A. S. Nair, P. A. Jeyanthy, L. Ramesh, G. M. Kurian and S. R. Mohamed, "Autonomous Precision Landing with UAV and Auto charging," 2021 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT), Bangalore, India, 2021, pp. 463-466, doi: 10.1109/RTEICT52294.2021.9573947.

[59] A. Marut, K. Wojtowicz and K. Falkowski, "ArUco markers pose estimation in UAV landing aid system," 2019 IEEE 5th International Workshop on Metrology for AeroSpace (MetroAeroSpace), Turin, Italy, 2019, pp. 261-266, doi: 10.1109/MetroAeroSpace.2019.8869572.

[60] W. Chinsatitf and T. Saitoh, "Improvement of eye detection performance for inside-out camera," 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), Okayama, Japan, 2016, pp. 1-6, doi: 10.1109/ICIS.2016.7550794.

[61] X. Ying and H. Zha, "Geometric Interpretations of the Relation between the Image of the Absolute Conic and Sphere Images," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 12, pp. 2031-2036, Dec. 2006, doi: 10.1109/TPAMI.2006.245.

[62] M. A. Martinez, "USV Attitude Position Estimation by a Hovering UAV Using Monocular Images of Deck-Mounted Lights," UC San Diego, 2022, ProQuest ID: Martinez_ucsd_0033M_21729. Merritt ID: ark:/13030/m5tr39jf, Retrieved from https://escholarship.org/uc/item/5p02s4st.

[63] J. Kennedy, I. Juricevic, Foreshortening gives way to forelengthening. Perception, 31, pp. 893-894, 2002.

[64] Y. Liu, Y. Dong, Z. Song and M. Wang, "2D-3D Point Set Registration Based on Global Rotation Search," in IEEE Transactions on Image Processing, vol. 28, no. 5, pp. 2599-2613, May 2019, doi: 10.1109/TIP.2018.2887207.

[65] S. Su et al., "A Novel Camera Calibration Method Based on Multilevel-Edge-Fitting Ellipse-Shaped Analytical Model," in IEEE Sensors Journal, vol. 20, no. 11, pp. 5818-5826, 1 June 1, 2020, doi: 10.1109/JSEN.2020.2972615.

[66] P. Sturm, "Pinhole Camera Model," In: Ikeuchi, K. (eds) Computer Vision. Springer, Boston, MA., 2014, https://doi.org/10.1007/978-0-387-31439-6_472

[67] E. Weisstein, "Ellipse," From MathWorld–A Wolfram Web Resource. https://mathworld.wolfram.com/Ellipse.html

[68] B. Peng, S. Dang and M. K. Dutta, "Feature detection and parameters calculation of arbitrary ellipse based on image processing technology," 2015 2nd International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India, 2015, pp. 418-420, doi: 10.1109/SPIN.2015.7095169.

[69] Y. Soh, J. Bae, D. Kim and H. Kim, "A New Method for Ellipse Fitting in Satellite Images," 2009 Second International Conference on Intelligent Computation Technology and Automation, Changsha, China, 2009, pp. 502-506, doi: 10.1109/ICICTA.2009.128.

[70] X. Morin, E. Pottier, J. Saillard, C. Pasdeloup and C. Delhote, "Polarimetric Detection of Slowly Moving Targets Embedded in Ground Clutter," Progress in Electromagnetics Research-pier - PROG ELECTROMAGN RES, 16, pp. 1-33, 1997, doi: 10.2528/PIER95103100.

[71] E. Bernardes and S. Viollet, "Quaternion to Euler angles conversion: A direct, general and computationally efficient method," PLoS ONE, 17, 2022, 10.1371/journal.pone.0276302.

[72] G. Bradski and A. Kaehler, "Learning OpenCV: Computer vision with the OpenCV library," O'Reilly Media, Inc, 2008.

[73] J. L. Sanchez-Lopez, J. Pestana, S. Saripalli, and P. Campoy, "An Approach Toward Visual Autonomous Ship Board Landing of a VTOL UAV," J. Intell. Robotics Syst. 74, 1–2, April 2014, pp. 113–127, doi: https://doi.org/10.1007/s10846-013-9926-3.

[74] C. Zou, Z. Yang, D. He, Y. Wei, J. Li, A. Jia, J. Chen, Q. Zhao, Y. Li, J. Li, S. Yang, "Theory, technology and prospects of conventional and unconventional natural gas," Petroleum Exploration and Development, Volume 45, Issue 4, 2018, pp. 604-618, ISSN 1876-3804, https://doi.org/10.1016/S1876-3804(18)30066-1.

[75] S. Nešić, "Key issues related to modelling of internal corrosion of oil and gas pipelines – A review," Corrosion Science, Volume 49, Issue 12, 2007, pp. 4308-4338, ISSN 0010-938X, https://doi.org/10.1016/j.corsci.2007.06.006.

[76] R. Parkins, "A Review of Stress Corrosion Cracking of High Pressure Gas Pipelines." Paper presented at the CORROSION 2000, Orlando, Florida, March 2000.

[77] S. D. Nivethika, T. Nivethetha, P. Priyadharshini, V. T. Nithyasri, M. SenthilPandian and R. Sivaprasad, "Design and Development of Pipe inspection Snake Locomotion Robot," 2022 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS), Chennai, India, 2022, pp. 1-5, doi: 10.1109/ICPECTS56089.2022.10047028.

[78] C-W. Huang, C-H. Huang, Y. Hung, and C. Chang, "Sensing pipes of a nuclear power mechanism using low-cost snake robot," Advances in Mechanical Engineering, 2018, 10(6), doi:10.1177/1687814018781286.

[79] T. Kamegawa, T. Baba and A. Gofuku, "V-shift control for snake robot moving the inside of a pipe with helical rolling motion," 2011 IEEE International Symposium on Safety, Security, and Rescue Robotics, Kyoto, Japan, 2011, pp. 1-6, doi: 10.1109/SSRR.2011.6106789.

[80] S. Burkett, and H. Schempf, "Wireless Self-powered Visual and NDE Robotic Inspection System for Live Gas Distribution Mains," United States: N. p., 2006, Web, doi:10.2172/876616.

[81] T. Li, S. Ma, B. Li, M. Wang and Y. Wang, "Design of spring parameters for a screw drive in-pipe robot based on energy consumption model," Proceeding of the 11th World Congress on Intelligent Control and Automation, Shenyang, China, 2014, pp. 3292-3297, doi: 10.1109/WCICA.2014.7053260.

[82] H. Tourajizadeh and M. Rezaei, "Design and control of a steerable screw in-pipe inspection robot," 2016 4th International Conference on Robotics and Mechatronics (ICROM), Tehran, Iran, 2016, pp. 98-104, doi: 10.1109/ICRoM.2016.7886824.

[83] J. Min, Y. Liauw, P. Pratama, S. Kim, H. Kim, "Development and Controller Design of Wheeled- Type Pipe Inspection Robot," Proceedings of the 2014 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2014, 10.1109/ICACCI.2014.6968543.

[84] Y. Kwon, E. Jung, H. Lim, B. Yi, "Design of a reconfigurable indoor pipeline inspection robot", Control, Automation and Systems, 2007, ICCAS '07, International Conference on, pp. 712-716, doi: 10.1109/ICCAS.2007.4406991.

[85] J. Knedlová, O. Bílek, D. Sámek, P. Chalupa, "Design and construction of an inspection robot for the sewage pipes," MATEC Web of Conferences, 2017, doi: 10.1051/matecconf/20171210100610.1051/matecconf/201712101006.

[86] A. Kakogawa and S. Ma, "Design of a multilink-articulated wheeled inspection robot for winding pipelines: AIRo-II," 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea (South), 2016, pp. 2115-2121, doi: 10.1109/IROS.2016.7759332.

[87] S. Mondal, A. Nandi, I. Mallick, C. Ghosh and A. Giri, "Performance evaluation of brushless DC motor drive for three different types of MOSFET based DC-DC converters," 2017 Devices for Integrated Circuit (DevIC), Kalyani, India, 2017, pp. 589-593, doi: 10.1109/DEVIC.2017.8074019.

[88] K. D. Truong, A. K. L. Luu, N. P. Tong, V. T. Duong, H. H. Nguyen and T. T. Nguyen, "Design of Series Elastic Actuator Applied for Humanoid," 2020 International Conference on Advanced Mechatronic Systems (ICAMechS), Hanoi, Vietnam, 2020, pp. 23-28, doi: 10.1109/ICAMechS49982.2020.9310118.

[89] C. Lee, S. Kwak, J. Kwak, S. Oh, "Generalization of Series Elastic Actuator Configurations and Dynamic Behavior Comparison," Actuators 6, no. 3, 2017, https://doi.org/10.3390/act6030026.

[90] G. Pratt, "Legged robots at MIT: what's new since Raibert?," IEEE Robotics & Automation Magazine, vol. 7, no. 3, pp. 15 – 19, 2000.

[91] F. Sergi, D. Accoto, G. Carpino, N. L. Tagliamonte and E. Guglielmelli, "Design and characterization of a compact rotary Series Elastic Actuator for knee assistance during overground walking," 2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob), Rome, Italy, 2012, pp. 1931-1936, doi: 10.1109/BioRob.2012.6290271.

[92] E. J. Rouse, L. M. Mooney, E. C. Martinez-Villalpando and H. M. Herr, "Clutchable series-elastic actuator: Design of a robotic knee prosthesis for minimum energy consumption,"

2013 IEEE 13th International Conference on Rehabilitation Robotics (ICORR), Seattle, WA, USA, 2013, pp. 1-6, doi: 10.1109/ICORR.2013.6650383.

[93] A. Zibafar, S. Ghaffari and G. Vossoughi, "Achieving transparency in series elastic actuator of sharif lower limb exoskeleton using LLNF-NARX model," 2016 4th International Conference on Robotics and Mechatronics (ICROM), Tehran, Iran, 2016, pp. 398-403, doi: 10.1109/ICRoM.2016.7886771.

[94] G. A. Pratt and M. M. Williamson, "Series elastic actuators," Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots, Pittsburgh, PA, USA, 1995, pp. 399-406 vol.1, doi: 10.1109/IROS.1995.525827.

[95] X. Yang, W. Luo, H. Liu, S. Zhang and W. Ge, "Optimal Design for the Torsional Elastic Element Topology of a Rotary Series Elastic Actuator," 2022 7th Asia-Pacific Conference on Intelligent Robot Systems (ACIRS), Tianjin, China, 2022, pp. 13-17, doi: 10.1109/ACIRS55390.2022.9845652.

[96] D. W. Robinson, J. E. Pratt, D. J. Paluska and G. A. Pratt, "Series elastic actuator development for a biomimetic walking robot," 1999 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (Cat. No.99TH8399), Atlanta, GA, USA, 1999, pp. 561-568, doi: 10.1109/AIM.1999.803231.

[97] X. Qu, D. Cao, Q. Wang and Y. Li, "Design and Research of Flexible Joint with Variable Stiffness Based on Torsion Spring," 2019 IEEE International Conference on Mechatronics and Automation (ICMA), Tianjin, China, 2019, pp. 325-329, doi: 10.1109/ICMA.2019.8816529.

[98] E. Campbell et al., "Design of a low-cost series elastic actuator for multi-robot manipulation," 2011 IEEE International Conference on Robotics and Automation, 2011, pp. 5395–5400.

[99] T. Hoeltgebaum, R. Luft, J. Elisii and R. Vieira, "A Design Comparison between Coil Springs and Torsion Bars," SAE Technical Paper 2012-36-0172, 2012, doi: 10.4271/2012-36-0172.

[100] An Sei Metaltek Company (n.d.). Torsion Spring Key Parameters and Reference Numbers. Spring Engineers. Retrieved June 20, 2023, from https://www.springhouston.com/products/torsion-springs/key-parameters-/reference-numbers.html

[101] Formlabs, "Rigid 4000 Resin for Stiff, Strong, Engineering-Grade Prototypes," accessed 22 August 2023, https://formlabs-media.formlabs.com/datasheets/1801088-TDS-ENUS-0.pdf